The ~~Ultimate~~ Guide to HTTP Resource Prioritization

Robin Marx @programmingart

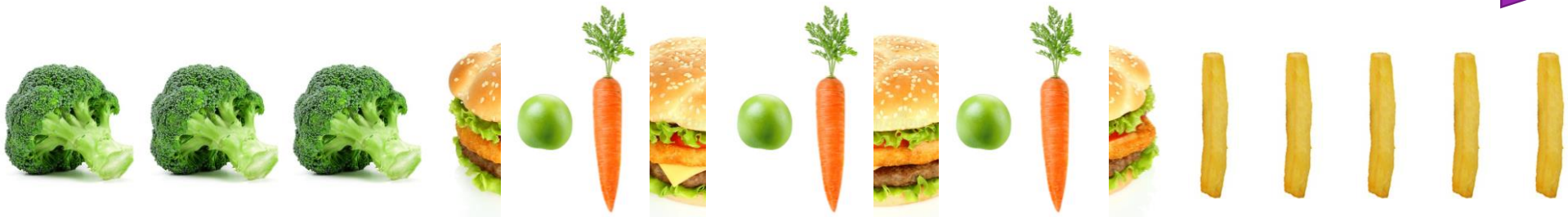
FOSDEM 2020


HELLO
LIPA
A Lipa City
Lifestyle Blog

A healthy, well-balanced meal

time

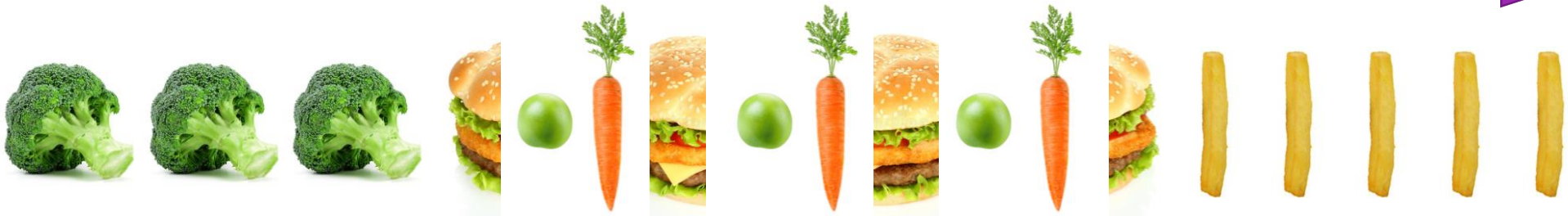
me



A healthy, well-balanced meal

time

me



girlfriend

A healthy, well-balanced meal




time

me

girlfriend

sister

A healthy, well-balanced meal

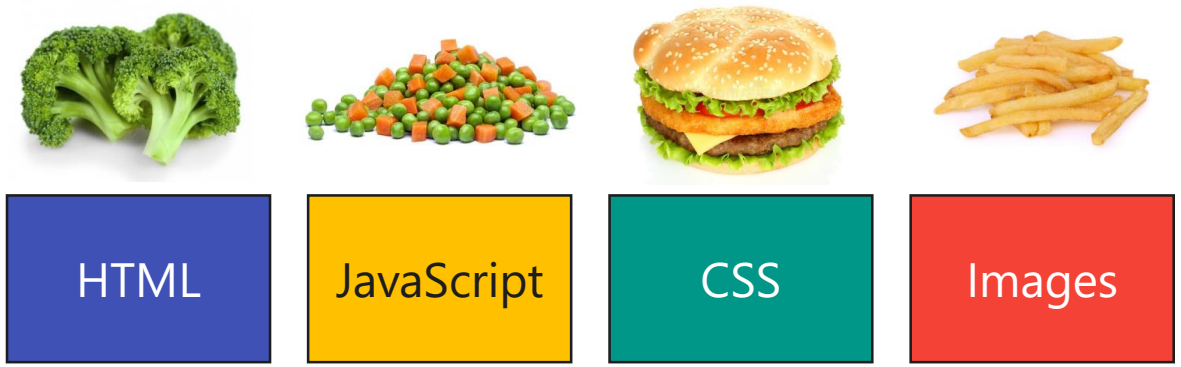
time

me

girlfriend

sister

dad
(aka: the lord of chaos)



HTTP/1.1 (TCP)

parallel

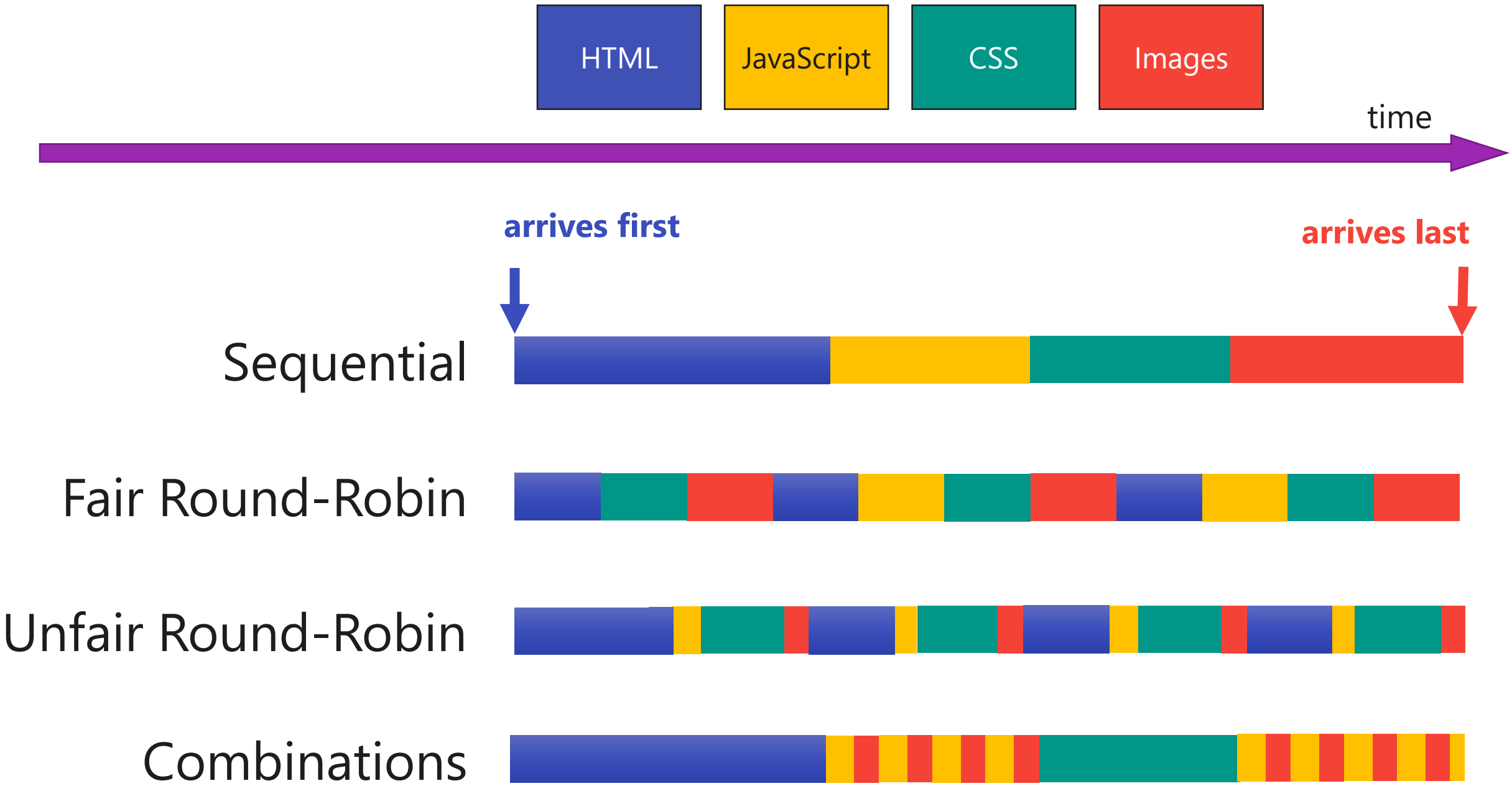
HTTP/2 (TCP)

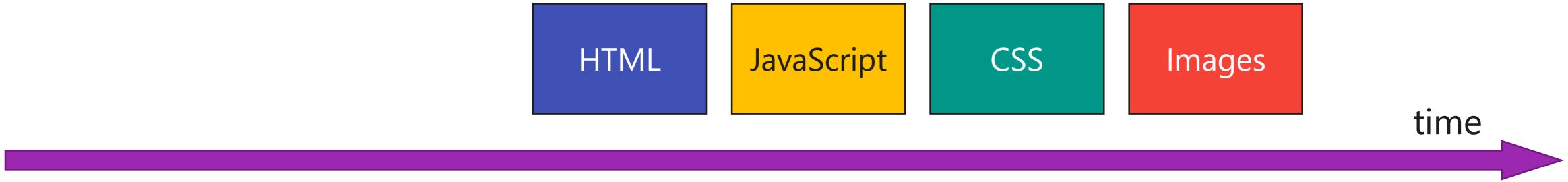
HTTP/3 (QUIC)

multiplexed

Problem 1:

What is the best multiplexing approach?

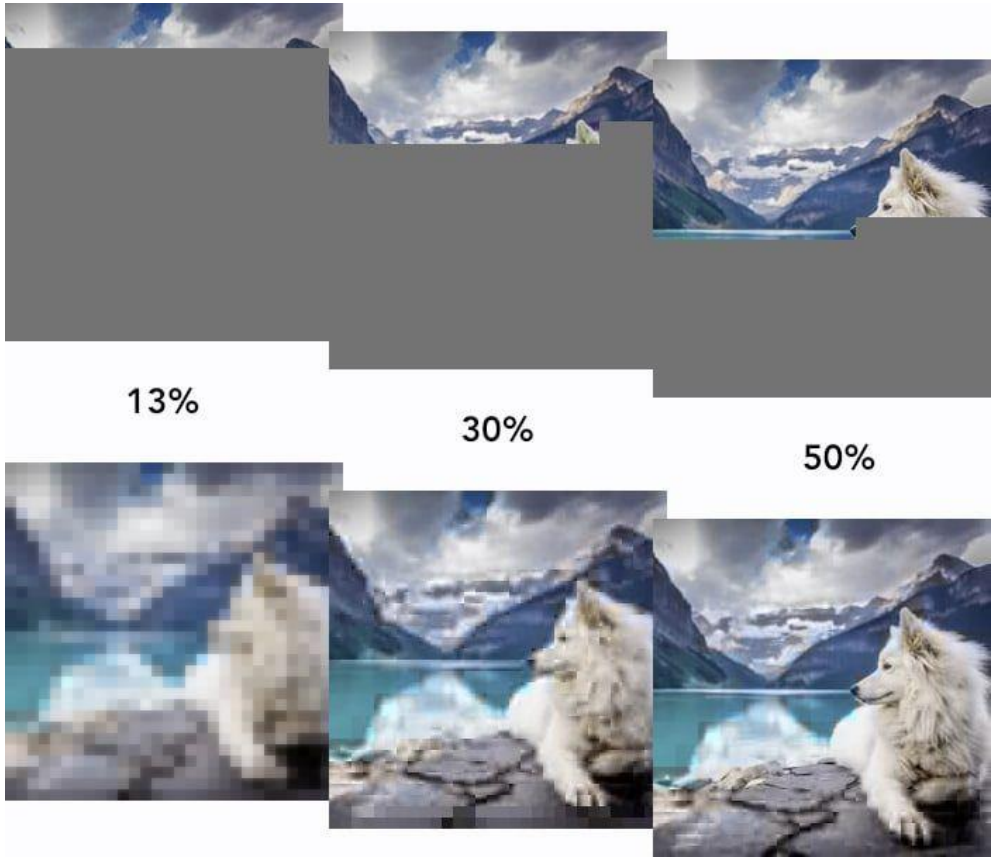

HTML
JavaScript
CSS
Images
time
arrives first
arrives last
Sequential
Fair Round-Robin
Unfair Round-Robin
Combinations




```
<head>
  <script src="script.js" />
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <img src="progressive1.jpg" />
  <img src="progressive2.jpg" />
</body>
```

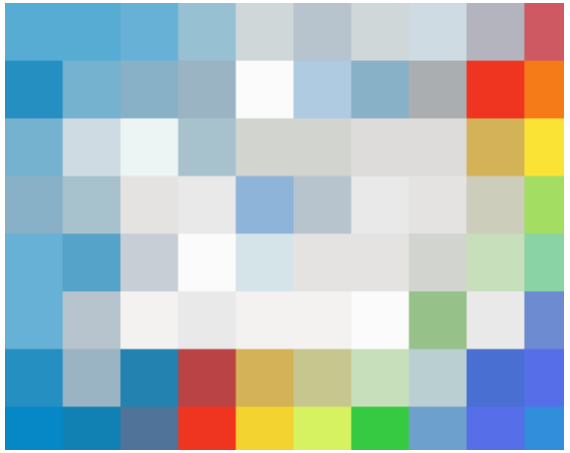
Render blocking

Progressive jpeg example

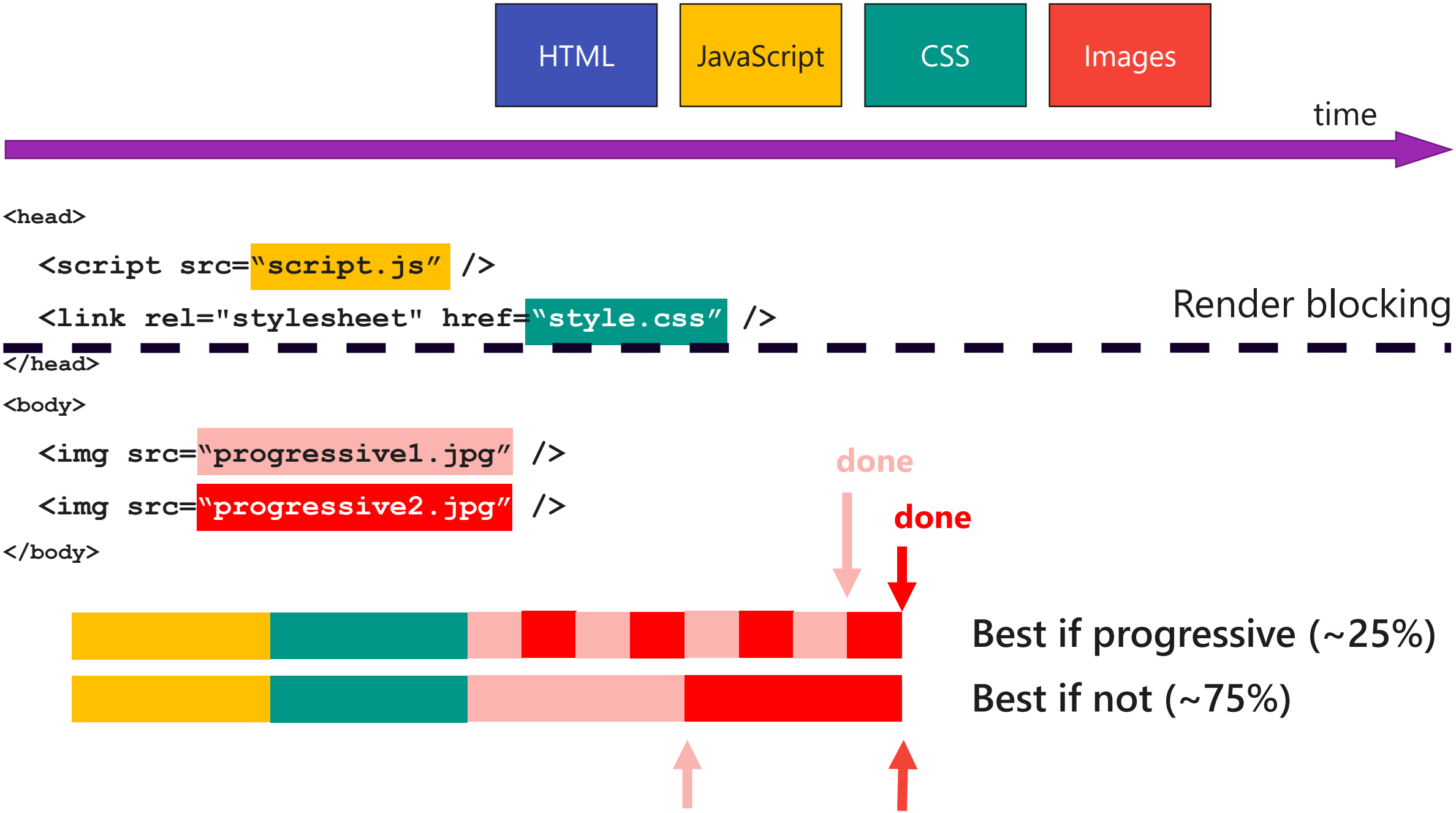


normal (scanline)

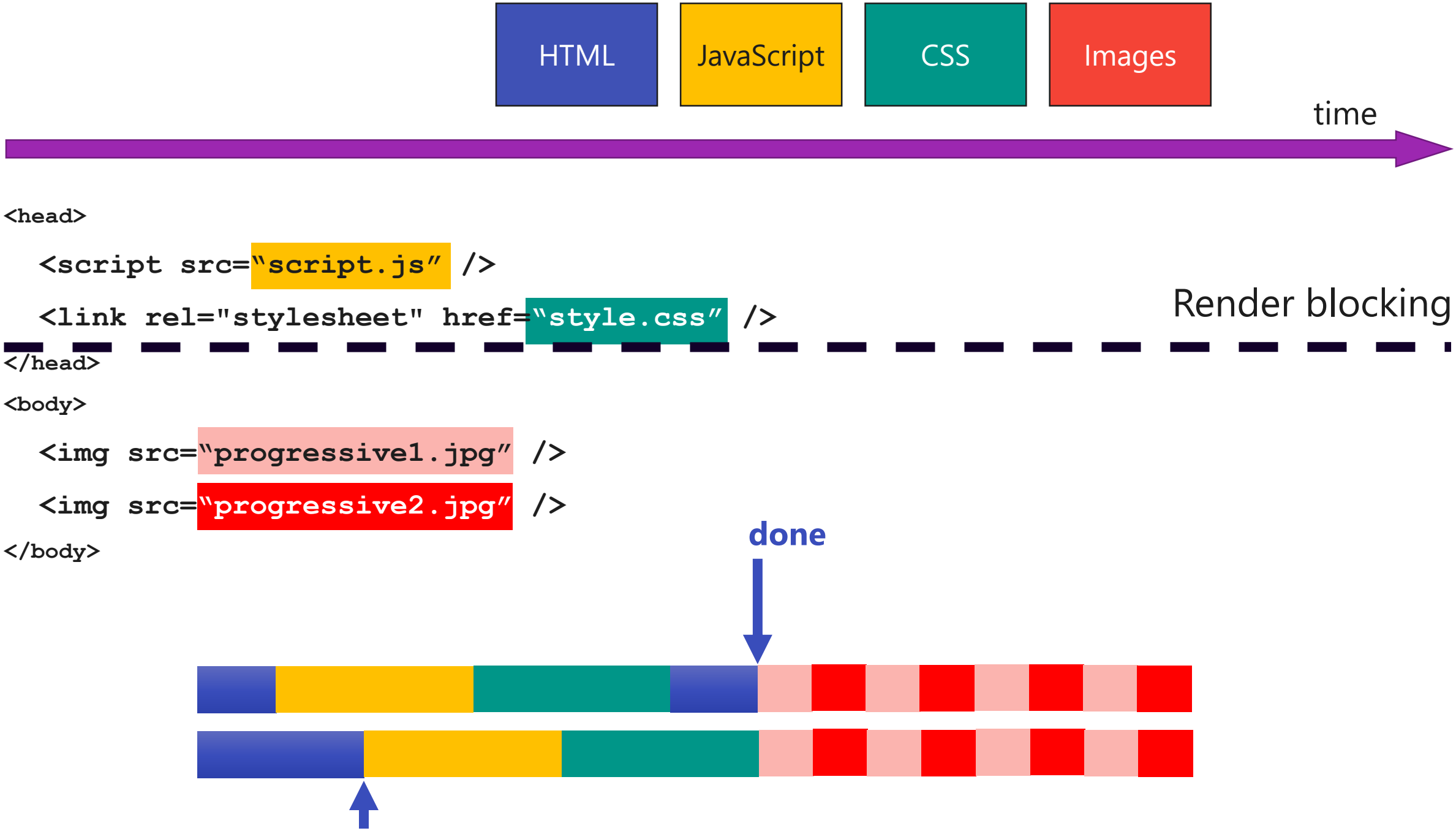
progressive



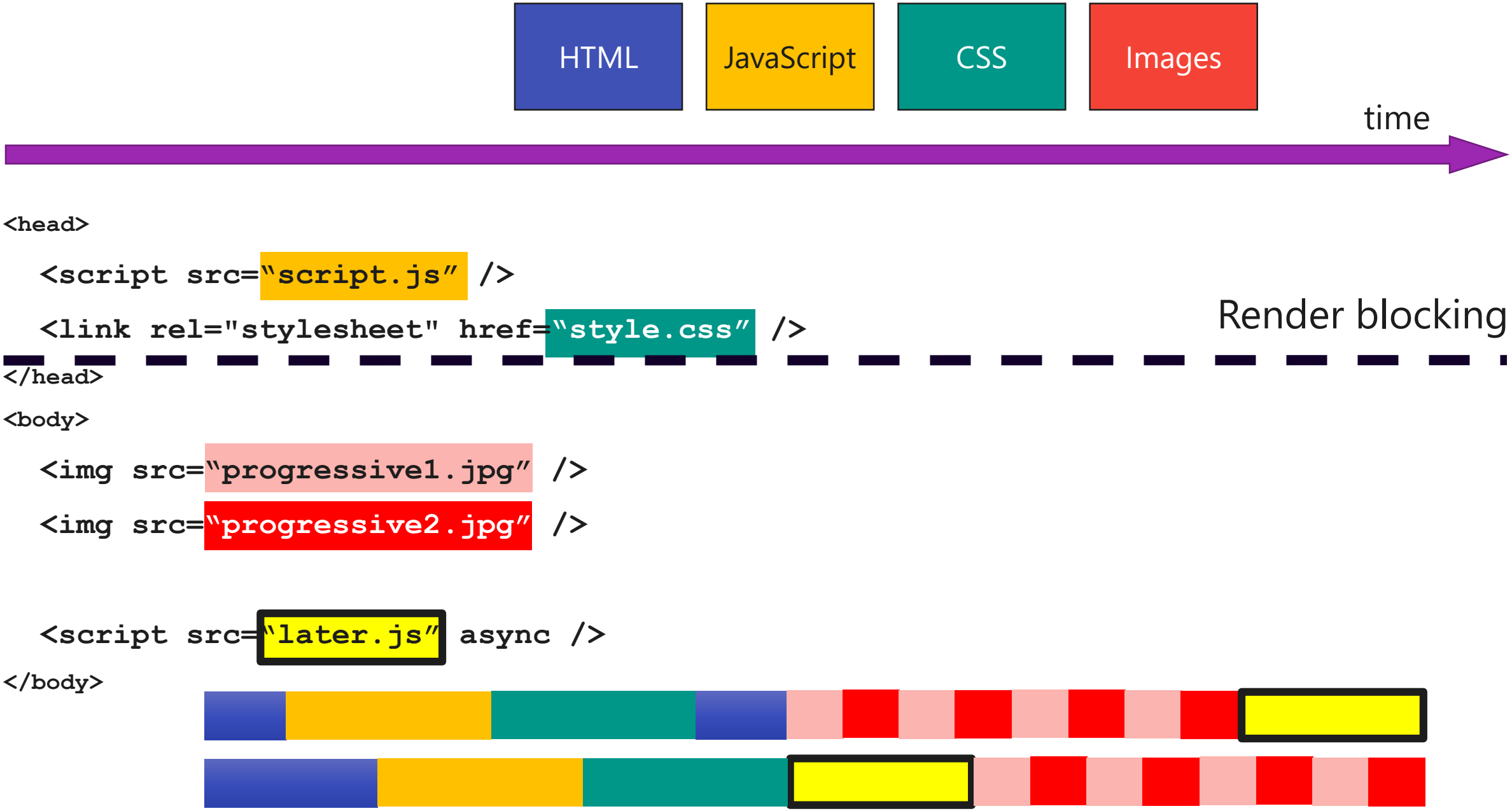
https://blog.cloudflare.com/parallel-streaming-of-progressive-images/
https://tobias.is/blogging/even-faster-images-using-http2-and-progressive-jpegs/


HTML
JavaScript
CSS
Images
time
<head>
<script src="script.js" />
<link rel="stylesheet" href="style.css" />
Render blocking
</head>
<body>

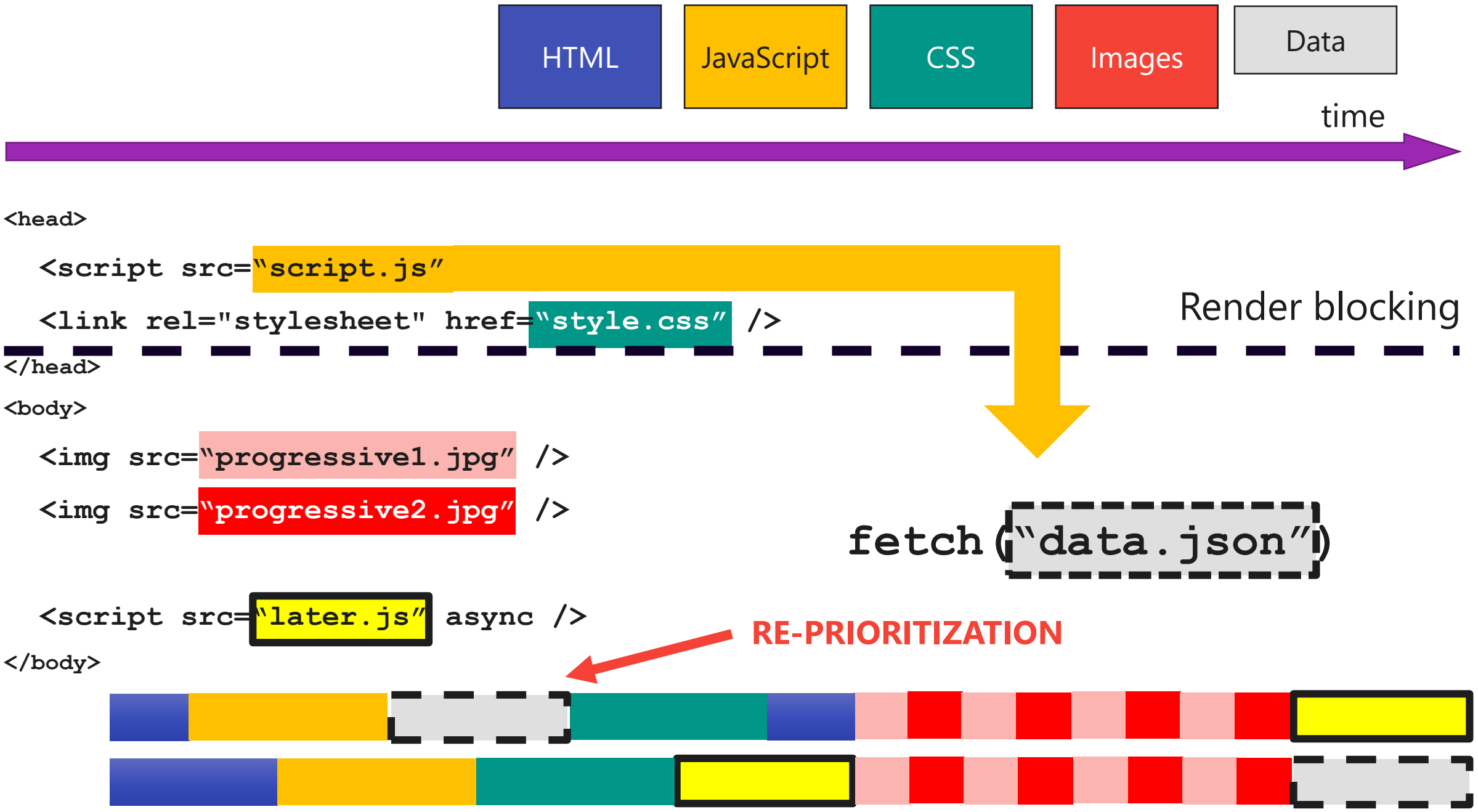
</body>
done
done
Best if progressive (~25%)
Best if not (~75%)
(~70 million jpegs in HTTP archive) https://twitter.com/programmingart/status/1222545046651785216?s=20


HTML
JavaScript
CSS
Images
time
<head>
<script src="script.js" />
<link rel="stylesheet" href="style.css" />
Render blocking
</head>
<body>

</body>
done


HTML
JavaScript
CSS
Images
time
<head>
<script src="script.js" />
<link rel="stylesheet" href="style.css" />
Render blocking
</head>
<body>

<script src="later.js" async />
</body>


HTML
JavaScript
CSS
Images
Data
time
<head>
<script src="script.js"
<link rel="stylesheet" href="style.css" />
Render blocking
</head>
<body>

fetch("data.json")
<script src="later.js" async />
RE-PRIORITIZATION
</body>

Browser doesn't know

1. Size of resource
2. If the resource can be used progressively
3. What the resource will actually do
4. If the resource references other resources
5. The "critical path"

http://web.mit.edu/polaris/
https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final177.pdf
https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-wang-xiao-sophia.pdf
https://hacks.mozilla.org/2017/09/building-the-dom-faster-speculative-parsing-async-defer-and-preload/

Browser doesn't know

1. Size of resource
2. If the resource can be used progressively
3. What the resource will actually do
4. If the resource references other resources
5. The "critical path"

So it has to guess

1. Mime-type
2. Position in the document
3. How it hopes developers use things like async, defer, preload, …

http://web.mit.edu/polaris/
https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final177.pdf
https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-wang-xiao-sophia.pdf
https://hacks.mozilla.org/2017/09/building-the-dom-faster-speculative-parsing-async-defer-and-preload/

Browser heuristics

HTML, CSS, **fonts**	HTML	HTML	
JS before img 1, **fetch**	JS, CSS	CSS, <head> JS	
visible img	**fonts**, **fetch**	**fonts**	
JS after img 1	img	img	
invisible img, async, defer		**fetch**, <body> JS	

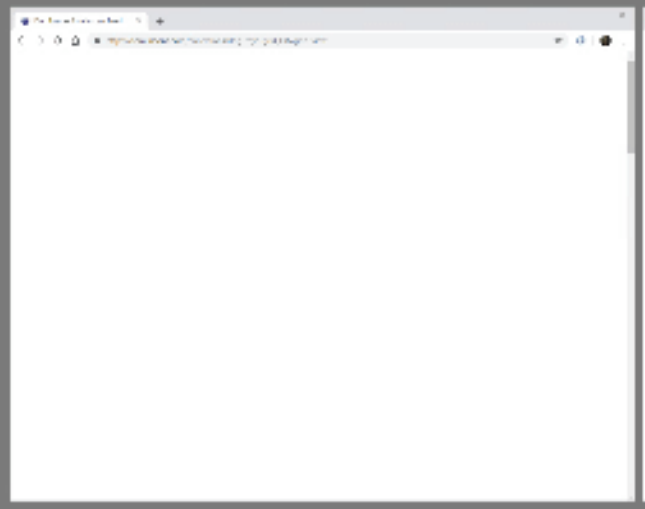
highest

lowest

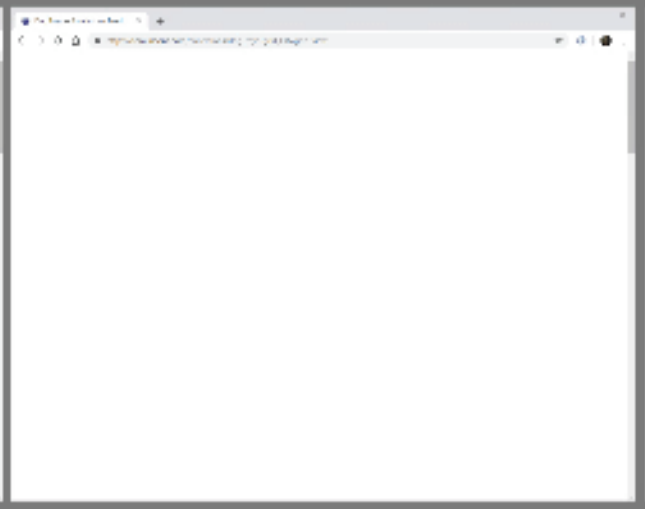
https://css-tricks.com/the-critical-request/
https://speeder.edm.uhasselt.be/www18/files/h2priorities_mwijnants_www2018.pdf
https://medium.com/reloading/preload-prefetch-and-priorities-in-chrome-776165961bbf

Which one is best?

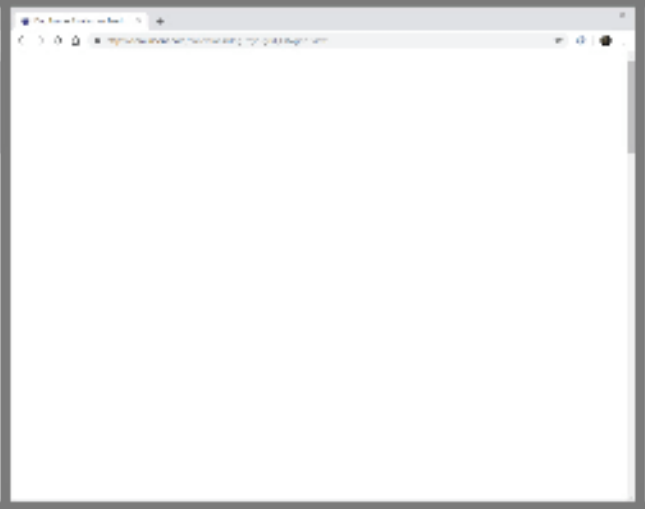
Edge



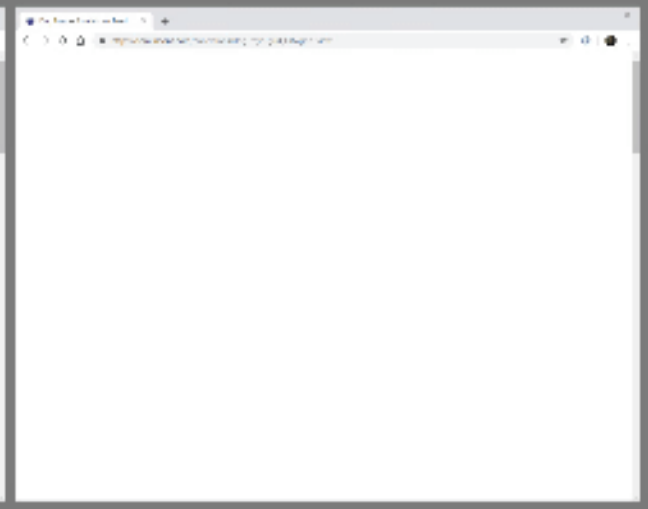
Safari

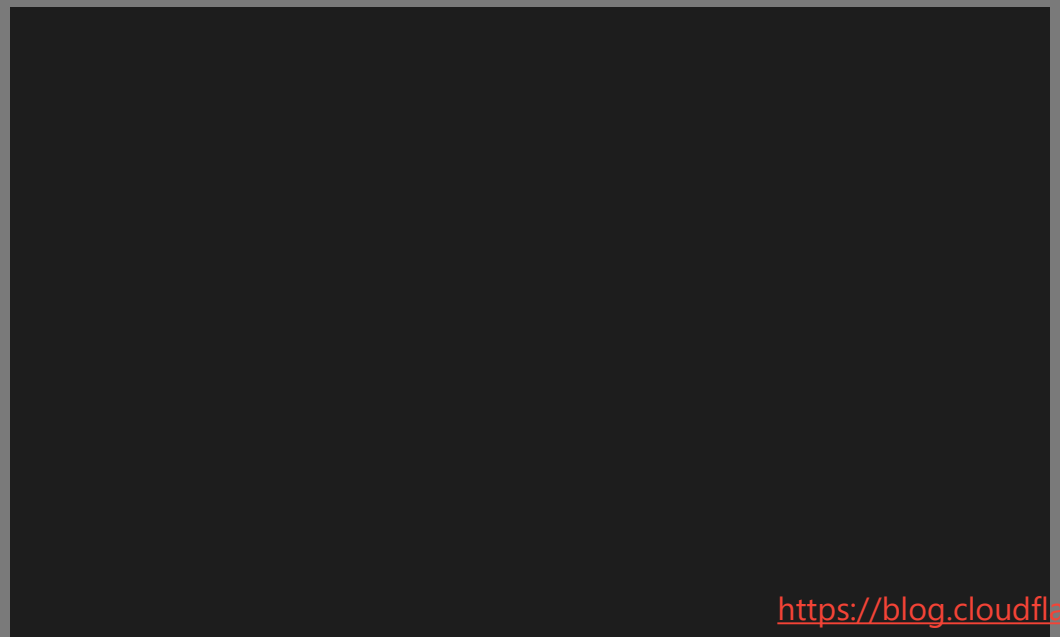


Firefox



Chrome





waiting...

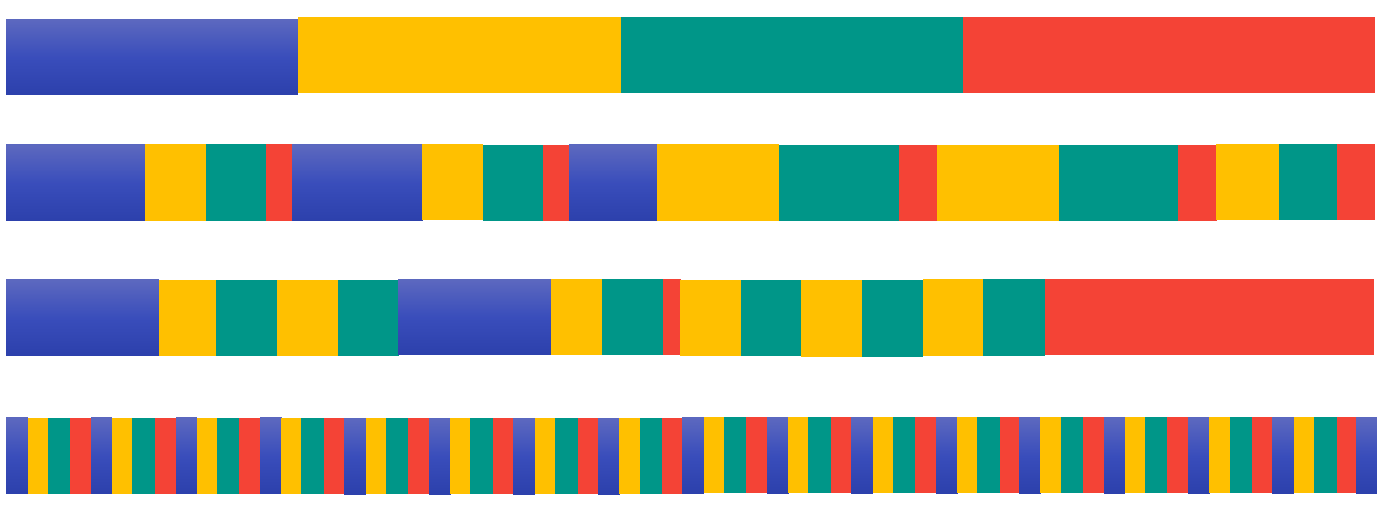
https://blog.cloudflare.com/better-http-2-prioritization-for-a-faster-web/

Browser heuristics

highest

lowest

HTML, CSS, **fonts**	HTML	HTML
JS before img 1, **fetch**	JS, CSS	CSS, <head> JS
visible img	**fonts**, **fetch**	**fonts**
JS after img 1	img	img
invisible img, async, defer		**fetch**, <body> JS



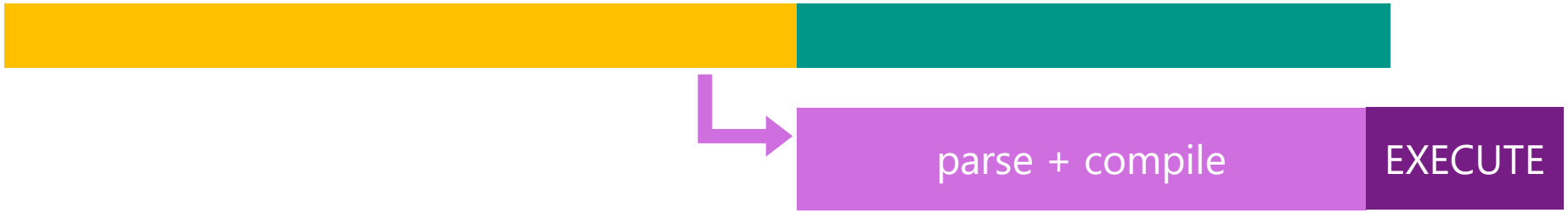
sequential

naïve unfair RR

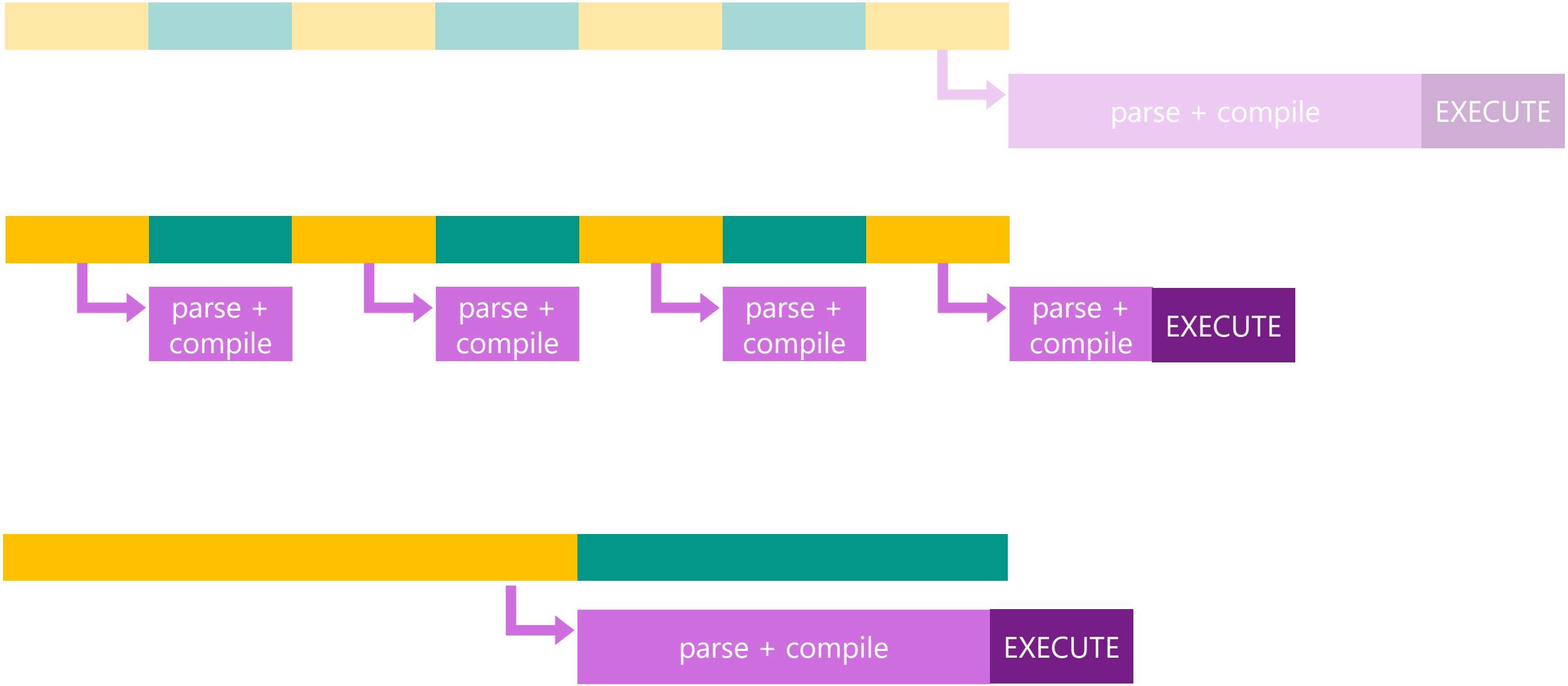
complex unfair RR

fair RR (H2 default)

Round-Robin is bad!

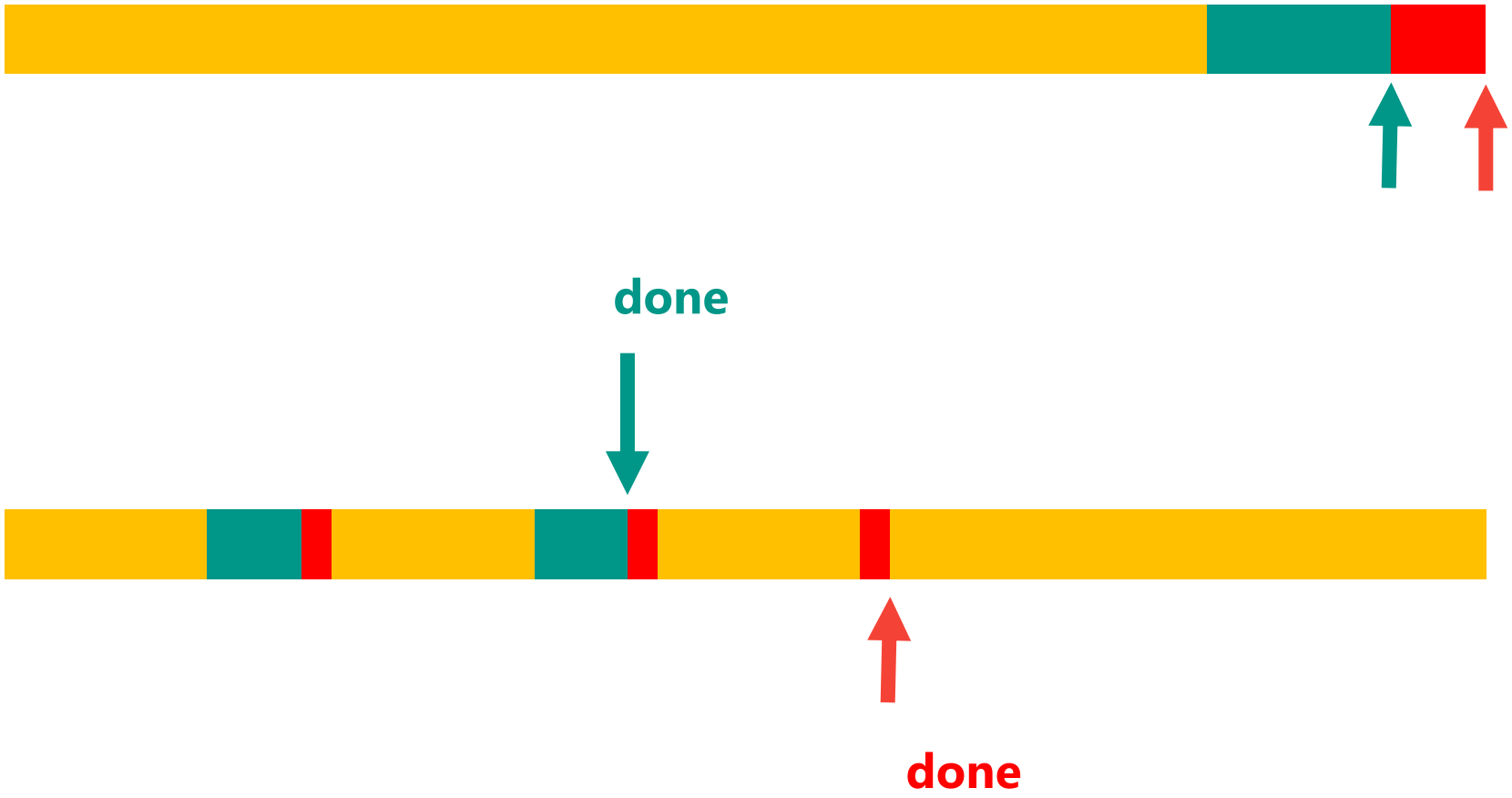


Round-Robin is bad!?

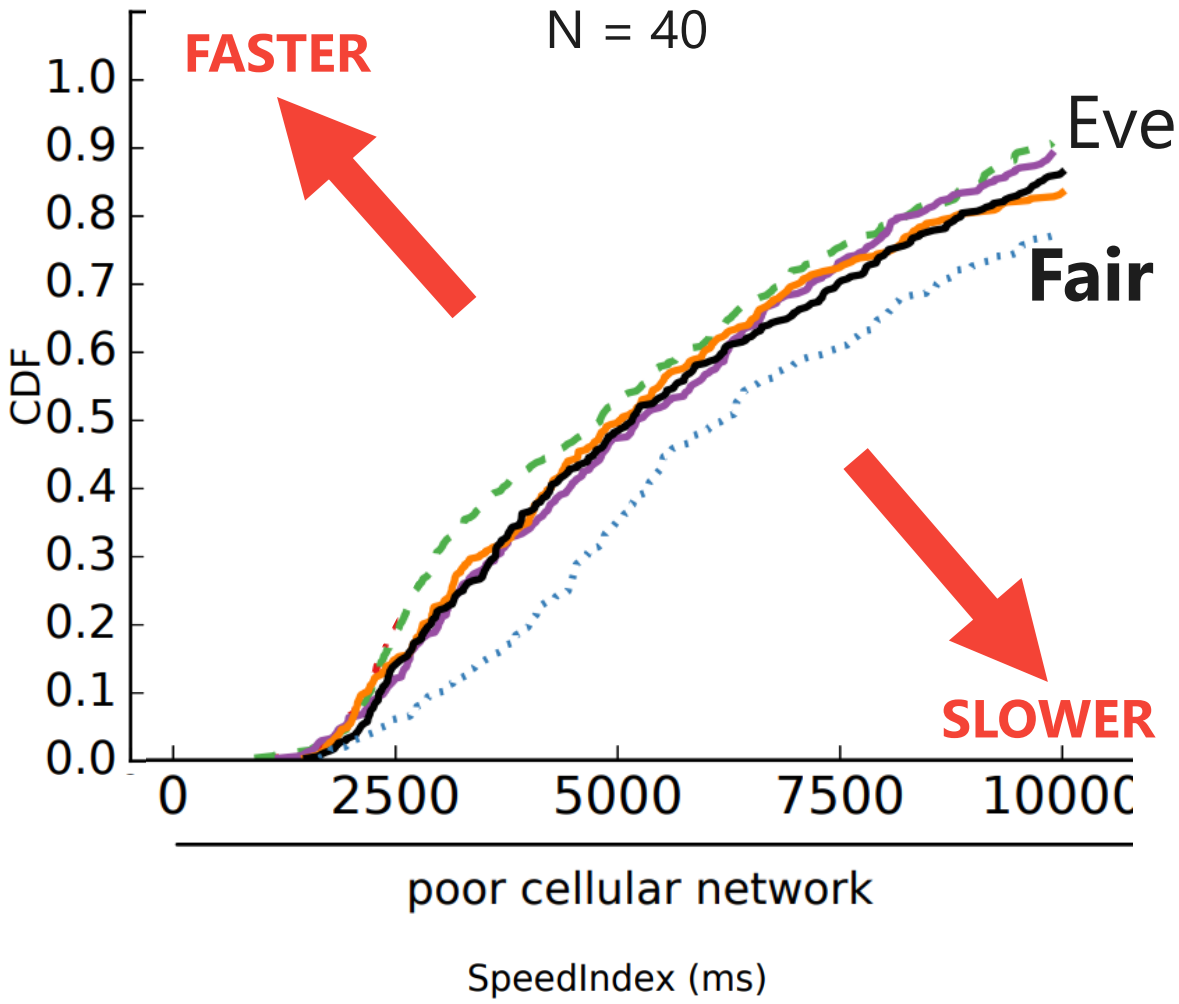


https://v8.dev/blog/v8-release-78
https://medium.com/reloading/javascript-start-up-performance-69200f43b201

Is Round-Robin bad?



Heuristics = on average

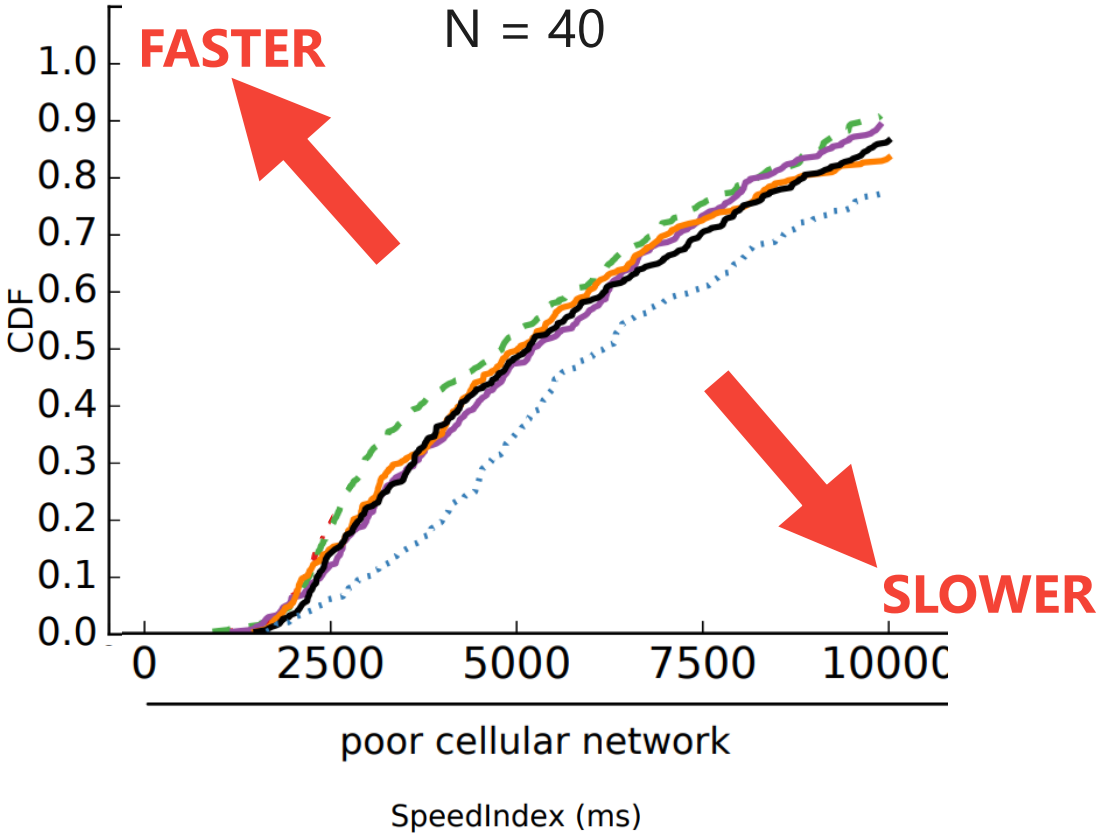


Everything else

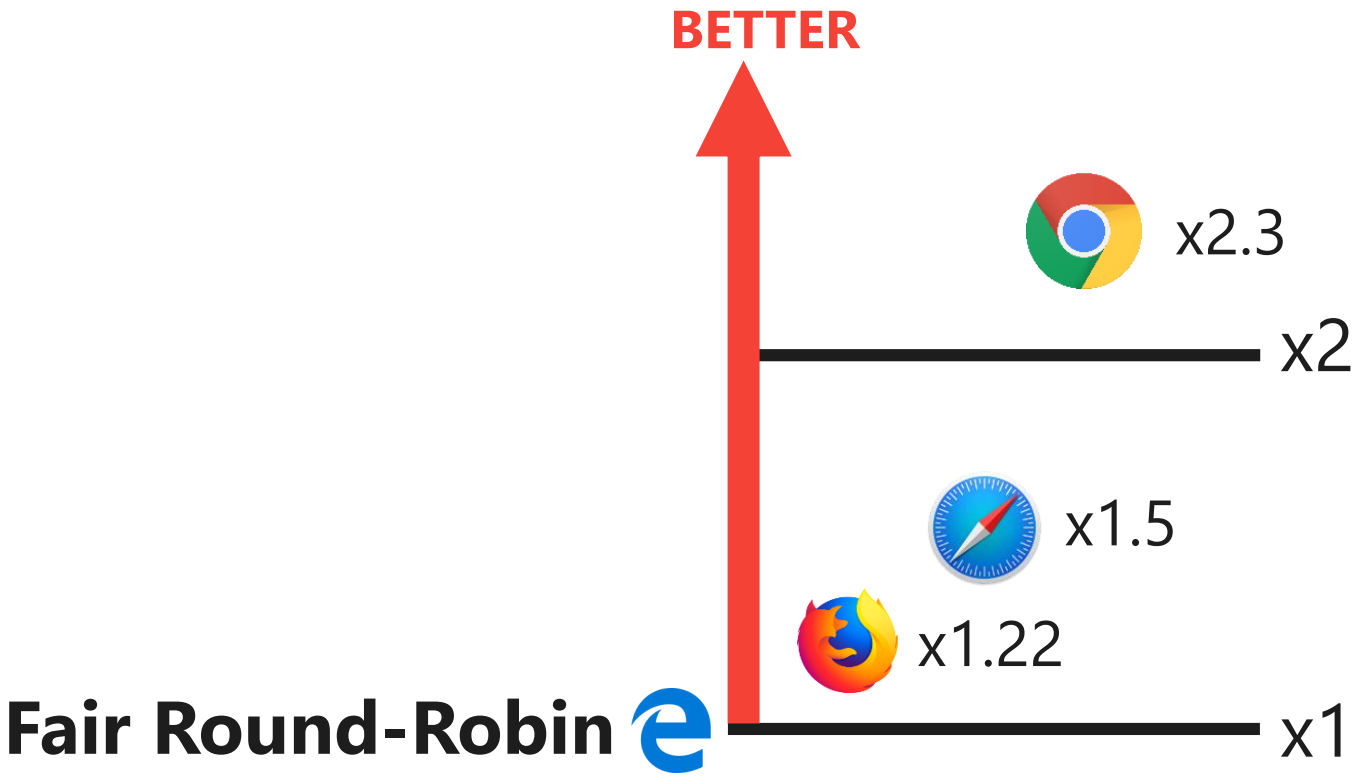
Fair Round-Robin

https://h3.edm.uhasselt.be/
https://speeder.edm.uhasselt.be/www18

Heuristics = on average

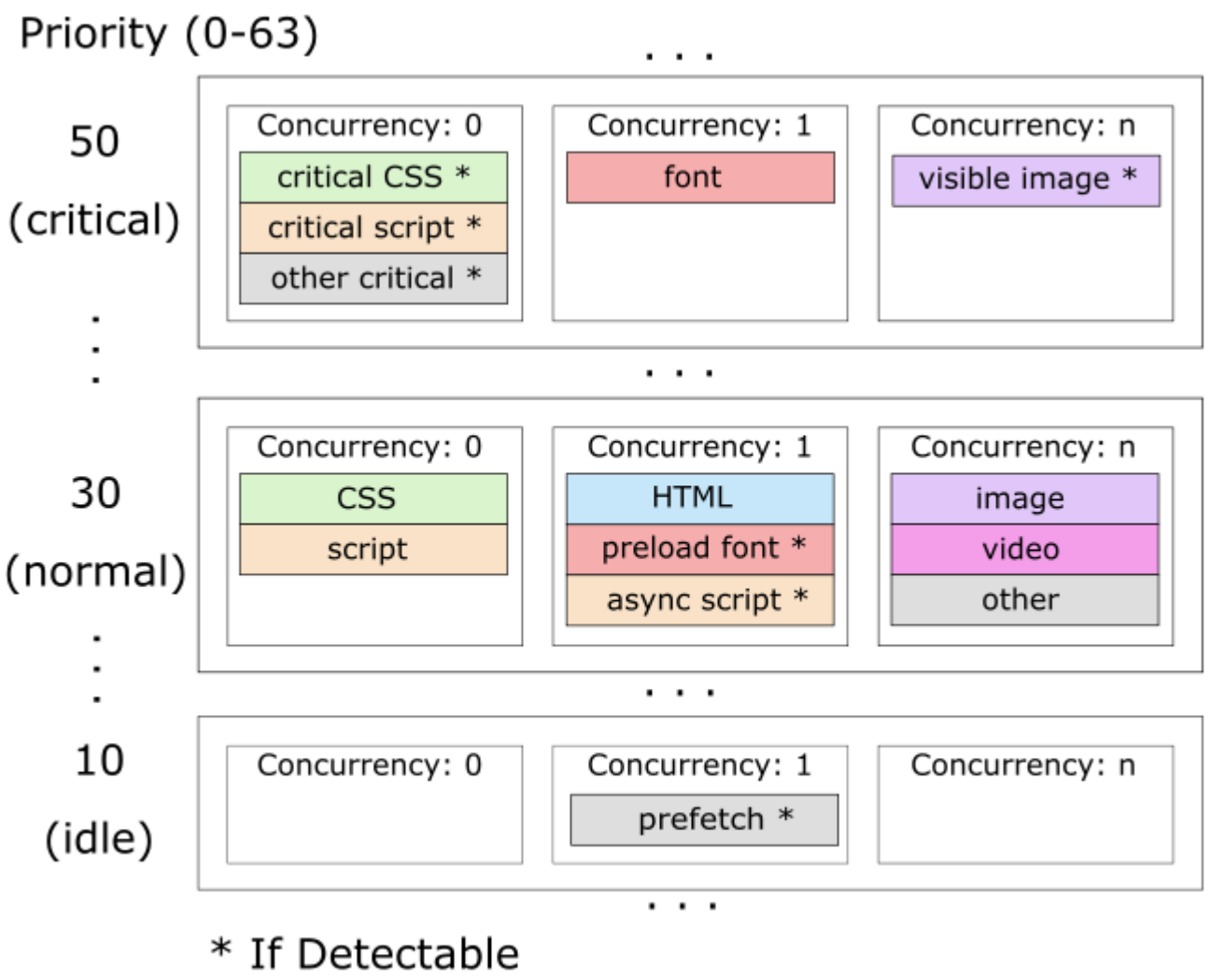


Average speedup factor for Above-the-fold resources



https://h3.edm.uhasselt.be/
https://speeder.edm.uhasselt.be/www18

Heuristics = on average



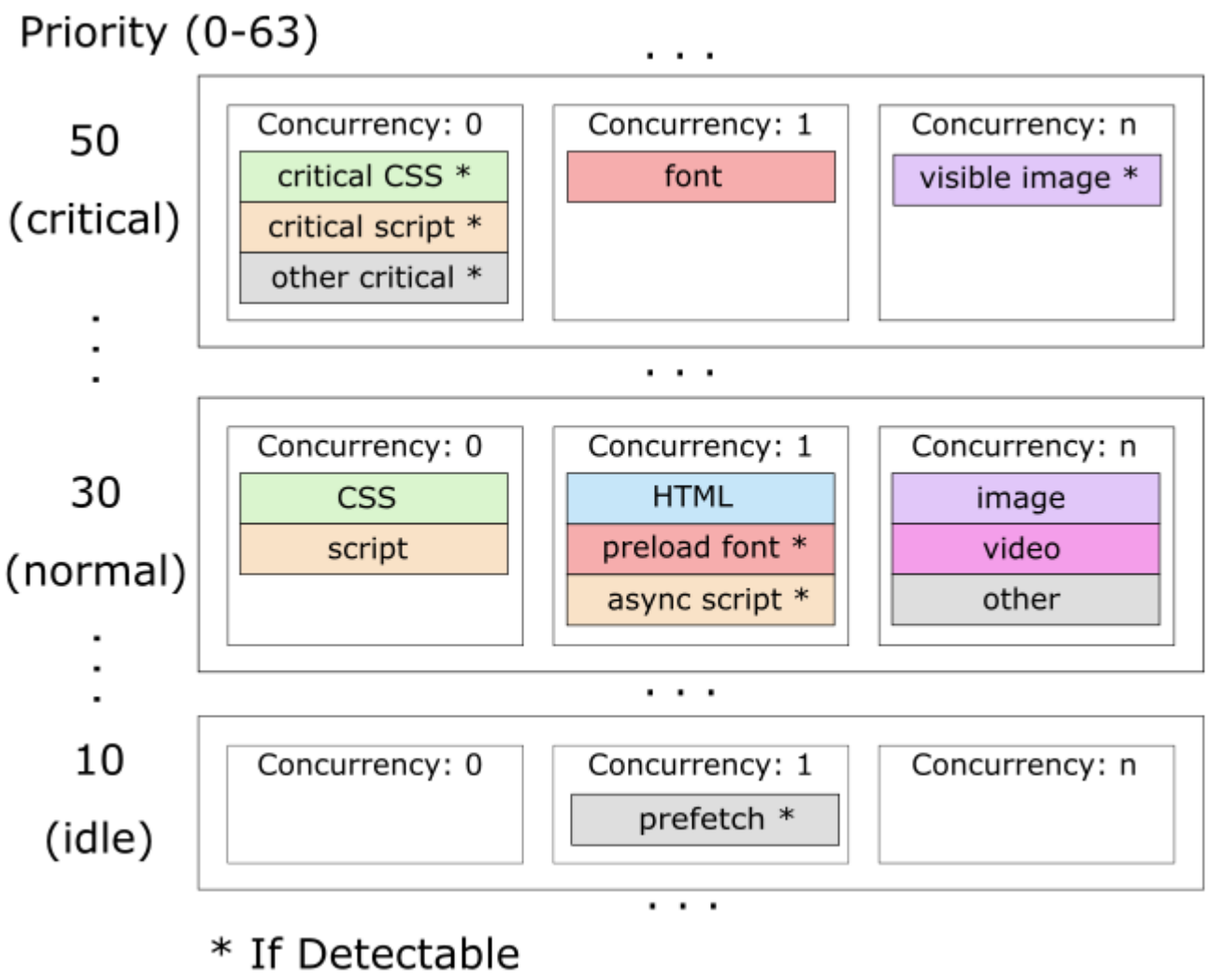
Sequential for CSS/Script

+

Complex Weighted RR for rest

https://blog.cloudflare.com/better-http-2-prioritization-for-a-faster-web/

Heuristics = on average



Sequential for CSS/Script

+

Complex Weighted RR for rest

“ 50% faster by default, particularly for **Edge** and **Safari** is not unusual ”

https://blog.cloudflare.com/better-http-2-prioritization-for-a-faster-web/

Heuristics = on average

N = 96

“The prioritizing scheduler beat the random scheduler on only 31% of pages tested”

2016

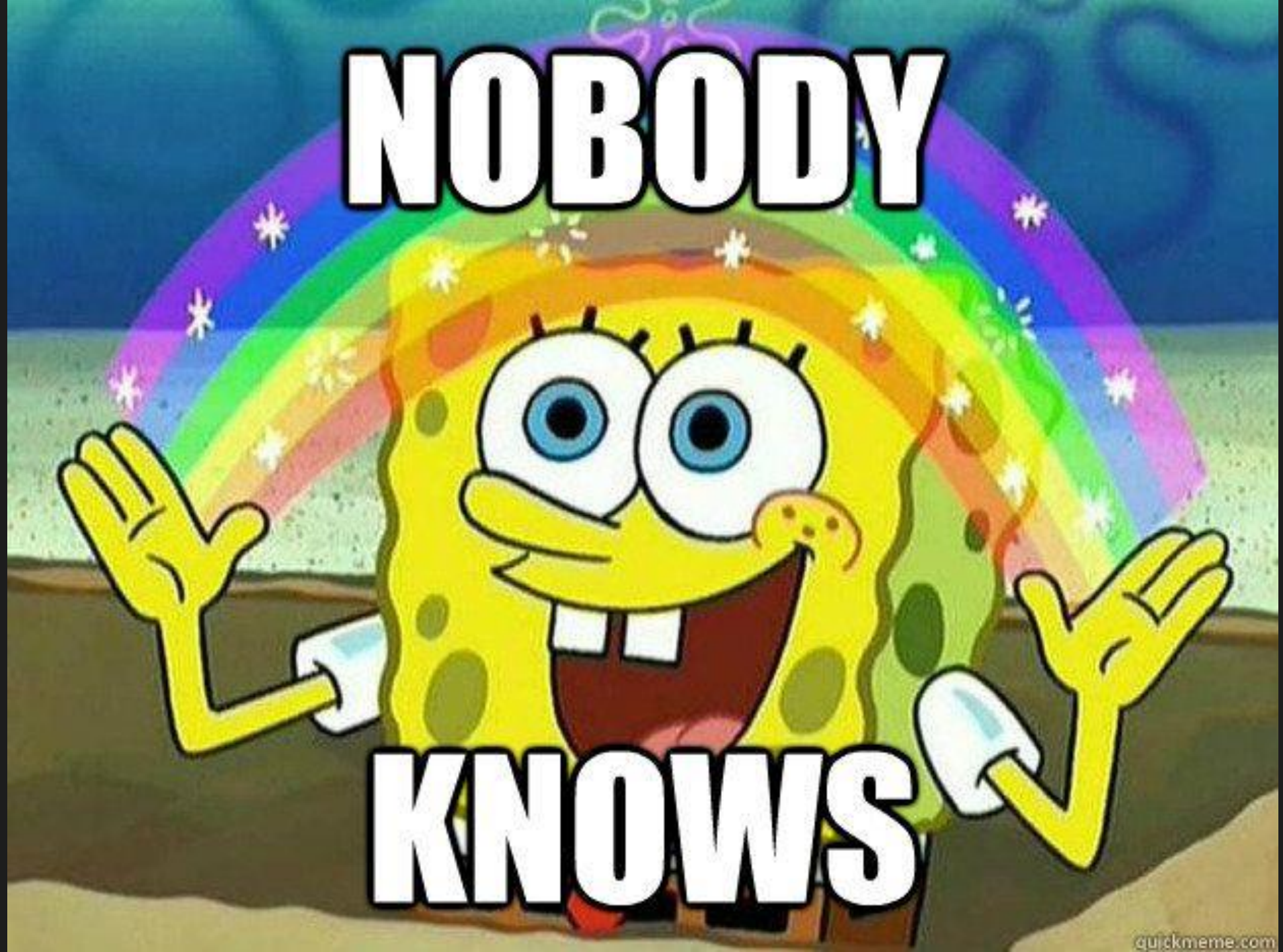
N = ?

“Chrome’s approach is better than fair RR, but only up to 2.69%”

“Maximum benefit was 3.1%, even compared to LIFO”

2019

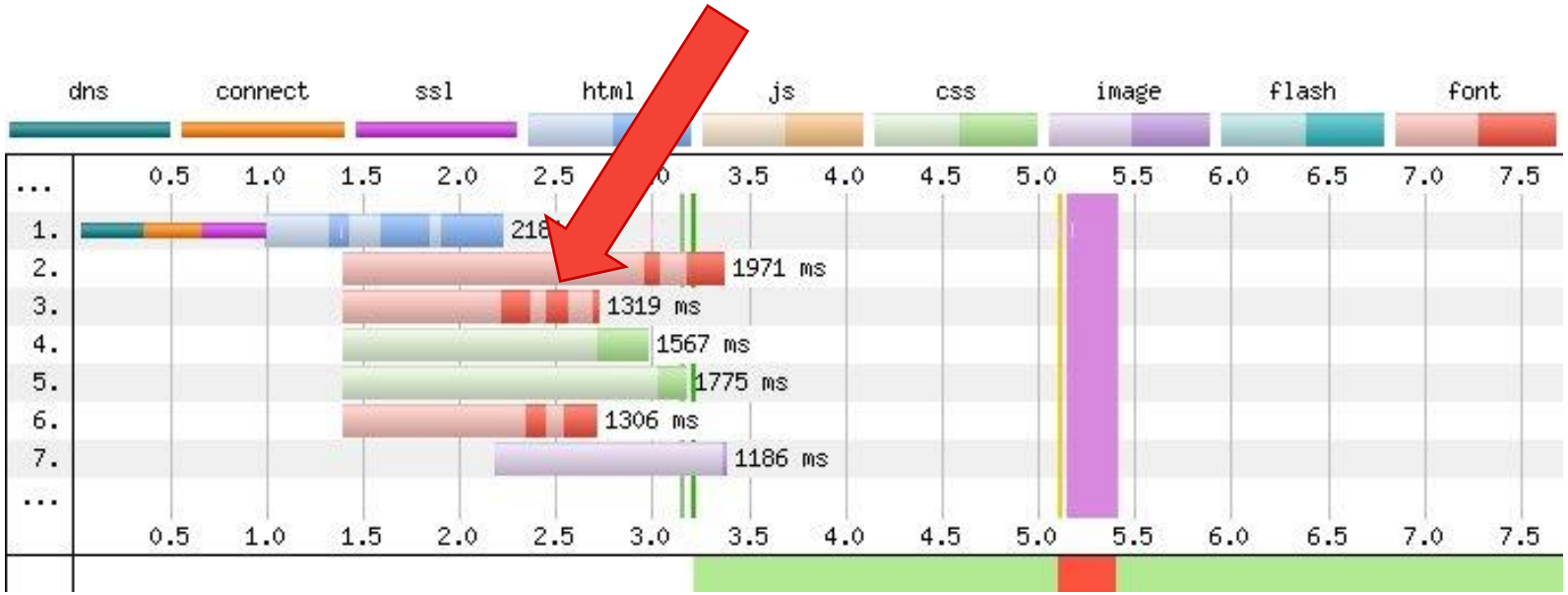
https://datatracker.ietf.org/meeting/106/materials/slides-106-httpbis-sessa-priorities-00.pdf
https://docs.google.com/document/d/1oLhNg1skaWD4_DtaoCxdSRN5erEXrH-KnLrMwEpOtFY


NOBODY
KNOWS
quickmeme.com

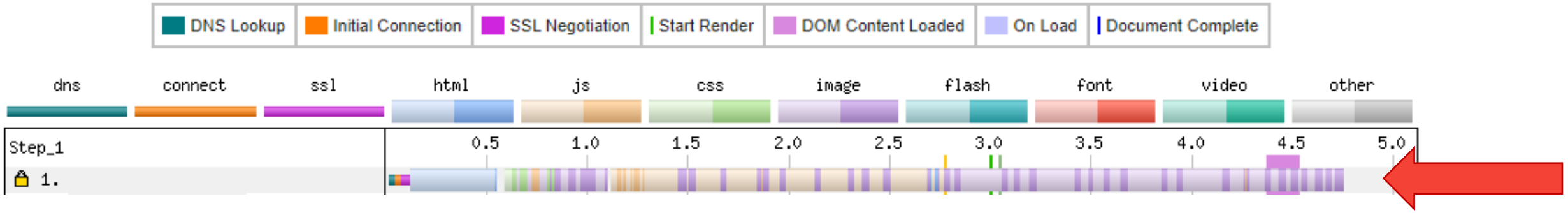
How can you find out if you have a problem?

Test your pages

webpagetest.org



Connection View



Some (imperfect) client-side options

1. Async, Defer
2. Preload

```
<link rel="preload" href="main.js" as="script">
```

```
<link rel="preload" href="style.css" as="style" onload="this.rel='stylesheet'">
```

But:
1. Bugs! (too aggressive)
2. Browser support 

https://wicg.github.io/priority-hints/
https://web.dev/native-lazy-loading/
https://bugzilla.mozilla.org/show_bug.cgi?id=1405761
https://twitter.com/domfarolino/status/1221803122638508032?s=20
https://andydavies.me/blog/2019/02/12/preloading-fonts-and-the-puzzle-of-priorities/

Some (imperfect) client-side options

1. Async, Defer
2. Preload

```
<link rel="preload" href="main.js" as="script">
```

```
<link rel="preload" href="style.css" as="style" onload="this.rel='stylesheet'">
```

But:
1. Bugs! (too aggressive)
2. Browser support 

3. Priority hints

```
<img src="img/carousel-1.jpg" importance="high">
```

```
fetch('/api/articles.json', { importance: 'high' }).then(/*...*/)
```

But:
1. Possibly not fine-grained enough
2. Browser support 

https://wicg.github.io/priority-hints/
https://web.dev/native-lazy-loading/
https://bugzilla.mozilla.org/show_bug.cgi?id=1405761
https://twitter.com/domfarolino/status/1221803122638508032?s=20
https://andydavies.me/blog/2019/02/12/preloading-fonts-and-the-puzzle-of-priorities/

Server-side overrides

https://www.shimmercat.com/blog/coordinated-image-loading.html
https://blog.cloudflare.com/parallel-streaming-of-progressive-images/
https://blog.cloudflare.com/better-http-2-prioritization-for-a-faster-web/
https://h2o.examp1e.net/configure/http2_directives.html#http2-reprioritize-blocking-assets

Problem 2:

How to communicate this to the server?

Please serve resources
in this order

origin

8 PRIORITY LEVELS

highest

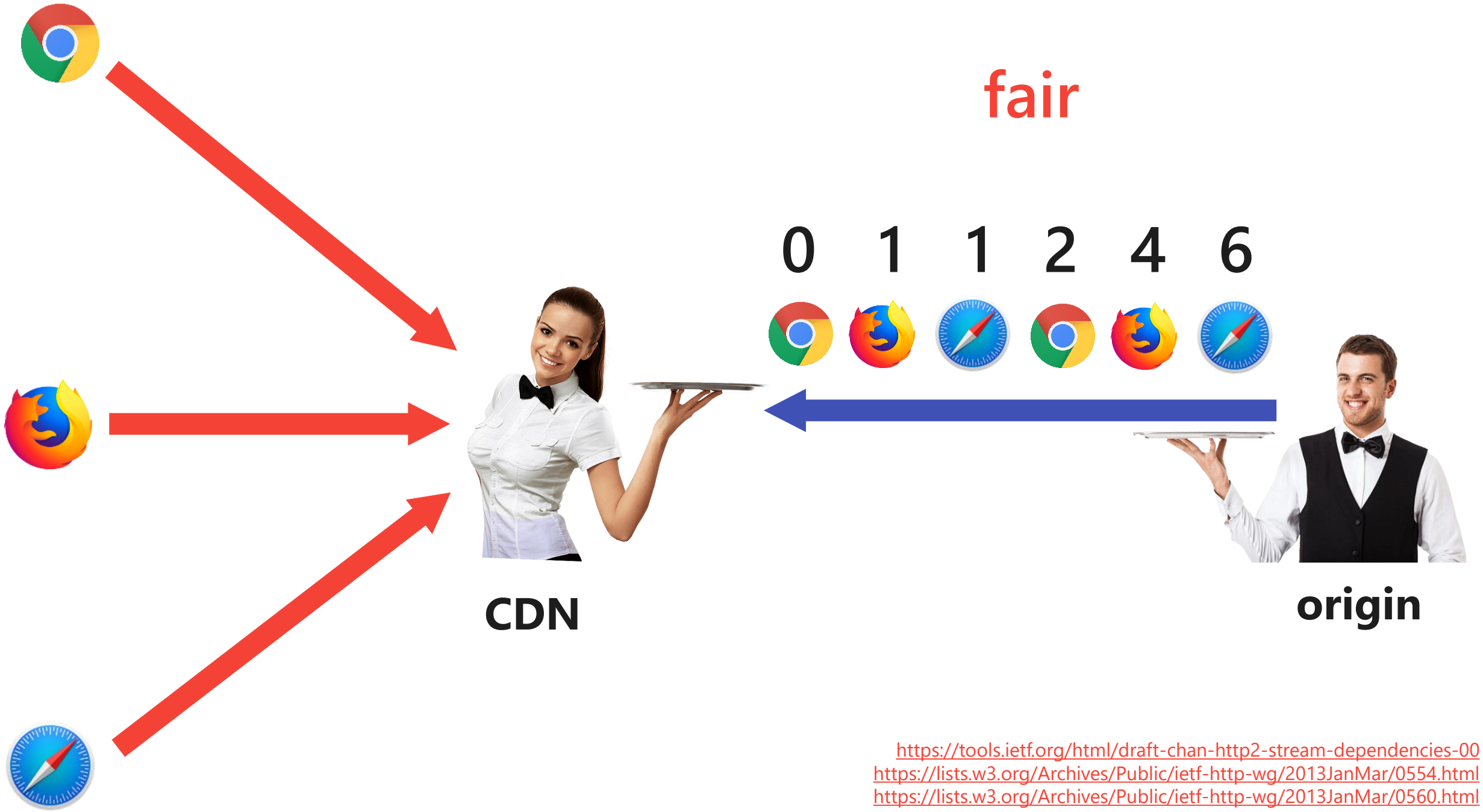
lowest

	Possible mapping
0	HTML
1	CSS
2	JavaScript
3	fonts
4	fetch
5	images
6	async and defer JS
7	video

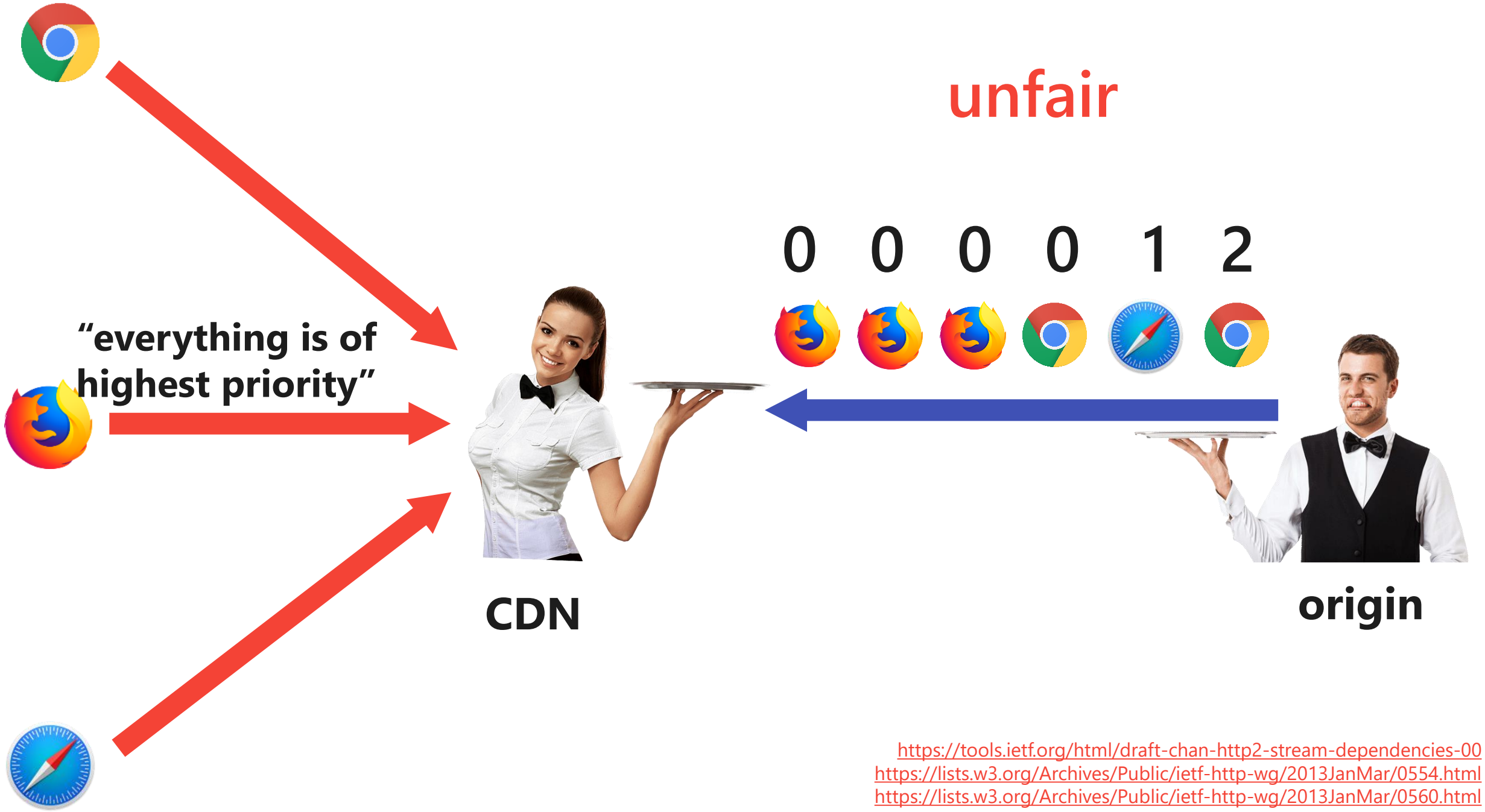
Google SPDY

origin

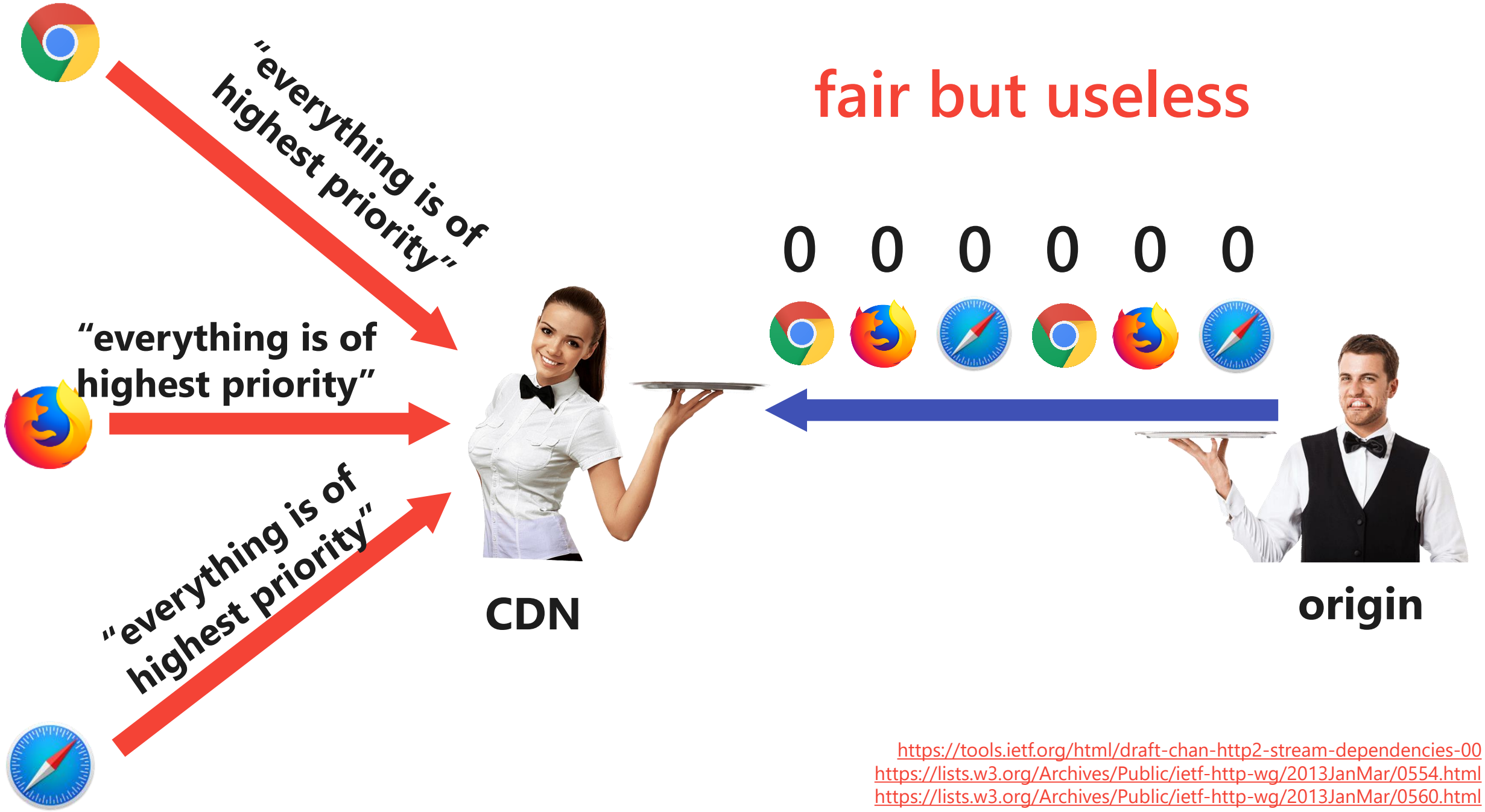
Please serve resources in this order



https://tools.ietf.org/html/draft-chan-http2-stream-dependencies-00
https://lists.w3.org/Archives/Public/ietf-http-wg/2013JanMar/0554.html
https://lists.w3.org/Archives/Public/ietf-http-wg/2013JanMar/0560.html
https://lists.w3.org/Archives/Public/ietf-http-wg/2019AprJun/0113.html

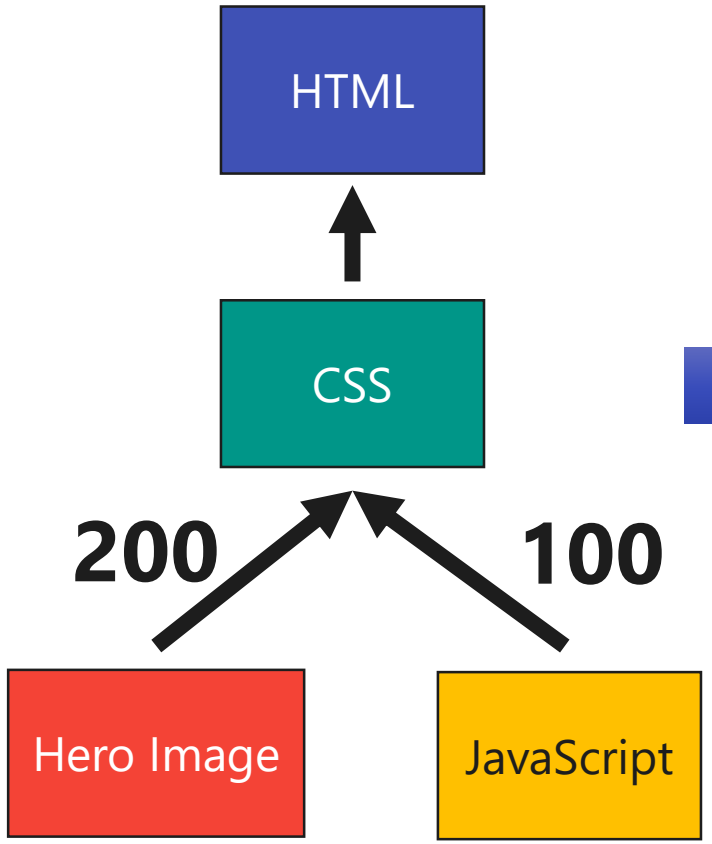


https://tools.ietf.org/html/draft-chan-http2-stream-dependencies-00
https://lists.w3.org/Archives/Public/ietf-http-wg/2013JanMar/0554.html
https://lists.w3.org/Archives/Public/ietf-http-wg/2013JanMar/0560.html
https://lists.w3.org/Archives/Public/ietf-http-wg/2019AprJun/0113.html



https://tools.ietf.org/html/draft-chan-http2-stream-dependencies-00
https://lists.w3.org/Archives/Public/ietf-http-wg/2013JanMar/0554.html
https://lists.w3.org/Archives/Public/ietf-http-wg/2013JanMar/0560.html
https://lists.w3.org/Archives/Public/ietf-http-wg/2019AprJun/0113.html

HTTP/2 : The Dependency Tree Awakens

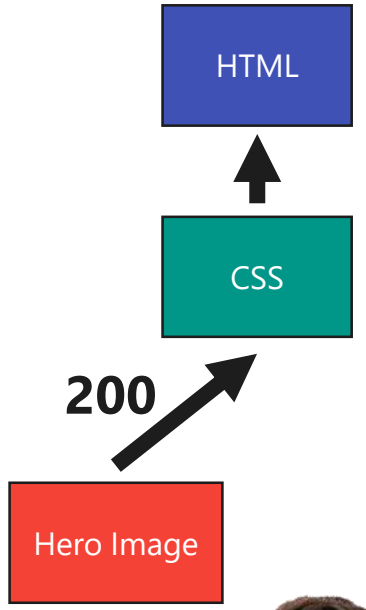


2:1 RR ratio

https://tools.ietf.org/html/draft-chan-http2-stream-dependencies-00

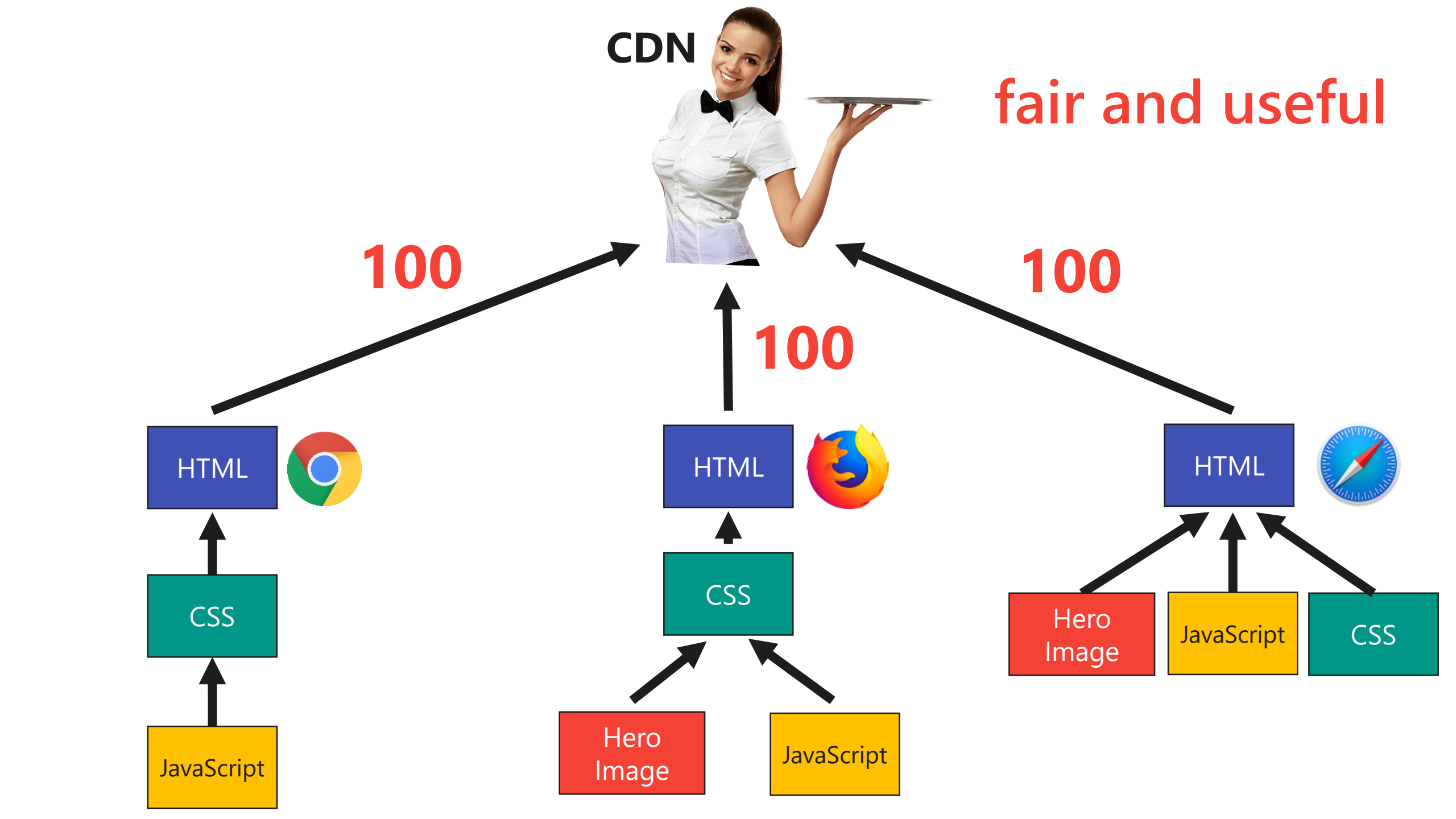
JavaScript

GET this file and add as child of CSS, with weight 100



origin

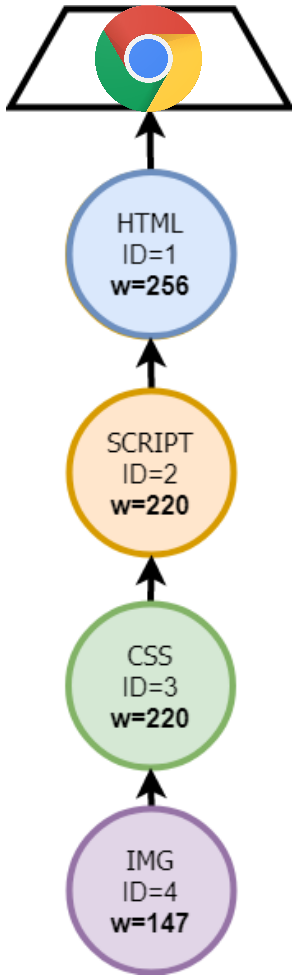
Please serve resources in this order

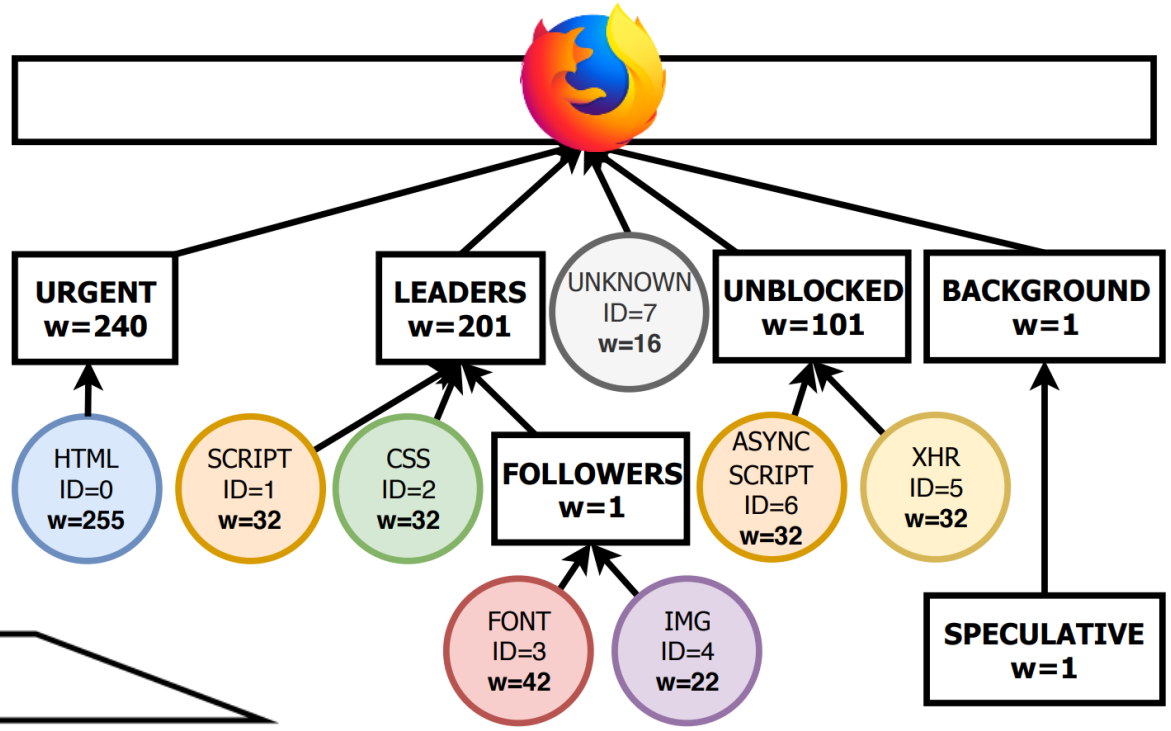

CDN
fair and useful
100
100
100
HTML
CSS
JavaScript
HTML
CSS
Hero Image
JavaScript
HTML
Hero Image
JavaScript
CSS

BUT: isn’t actually used that way

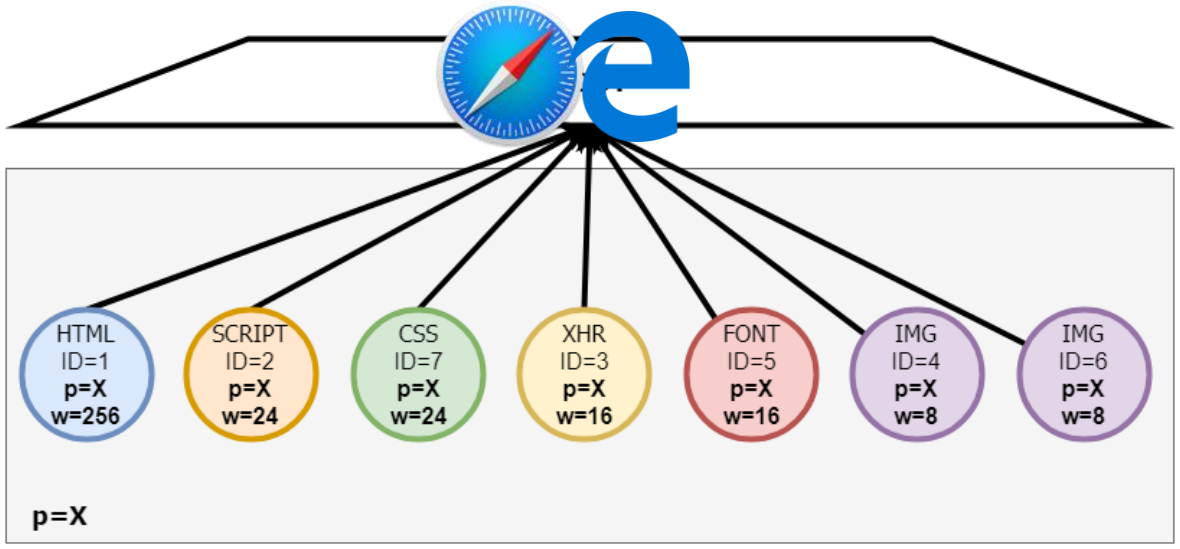
BUT: only firefox uses a real tree

No siblings





Only siblings

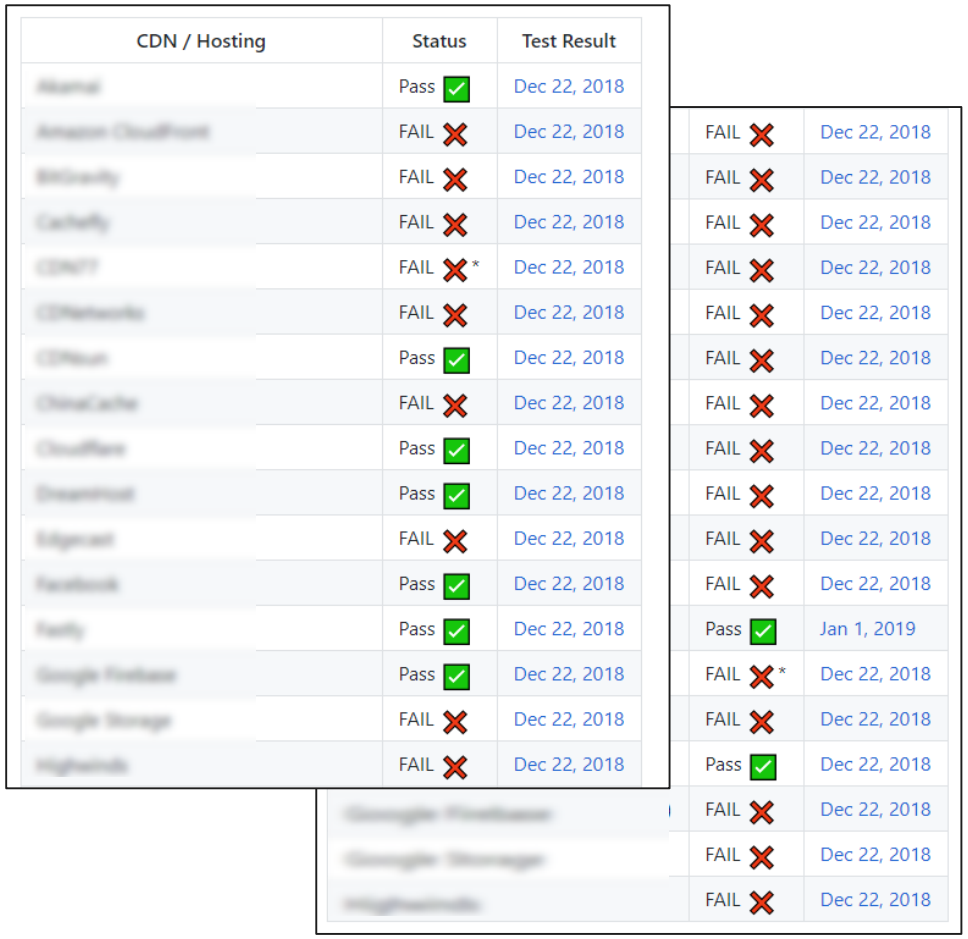


Siblings with ``placeholders"

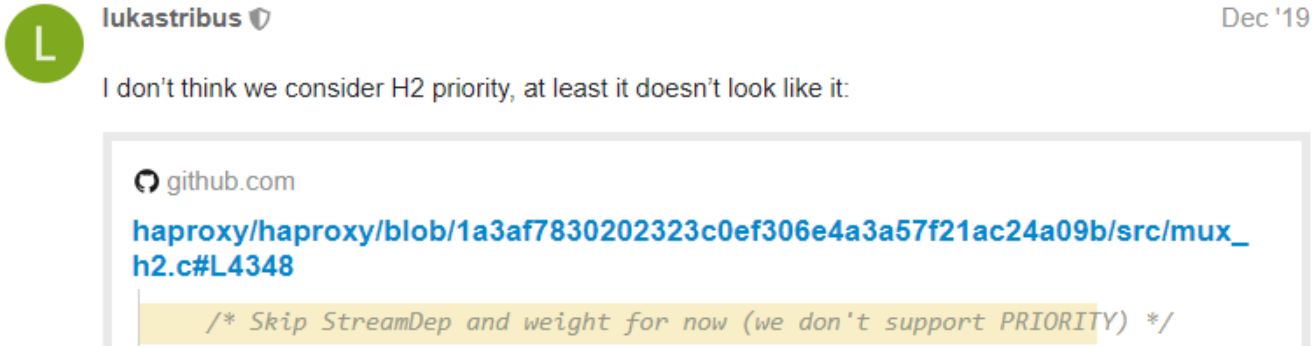
https://speeder.edm.uhasselt.be/www18

http://bitsup.blogspot.com/2015/01/http2-dependency-priorities-in-firefox.html

BUT: servers implement badly or not at all



CDN / Hosting	Status	Test Result
[illegible]	Pass	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL *	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	Pass	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	Pass	Dec 22, 2018
[illegible]	Pass	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	Pass	Dec 22, 2018
[illegible]	Pass	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	Pass	Jan 1, 2019
[illegible]	FAIL *	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	Pass	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018
[illegible]	FAIL	Dec 22, 2018



lukastribus Dec '19

I don't think we consider H2 priority, at least it doesn't look like it:

github.com

haproxy/haproxy/blob/1a3af7830202323c0ef306e4a3a57f21ac24a09b/src/mux_h2.c#L4348

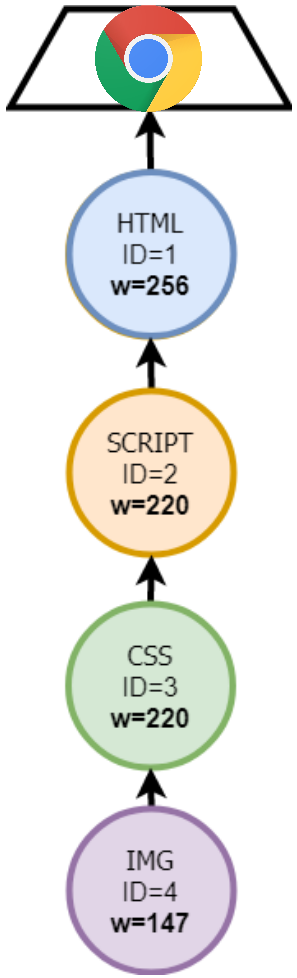
```
/* Skip StreamDep and weight for now (we don't support PRIORITY) */
```

9 / 34 deployments pass

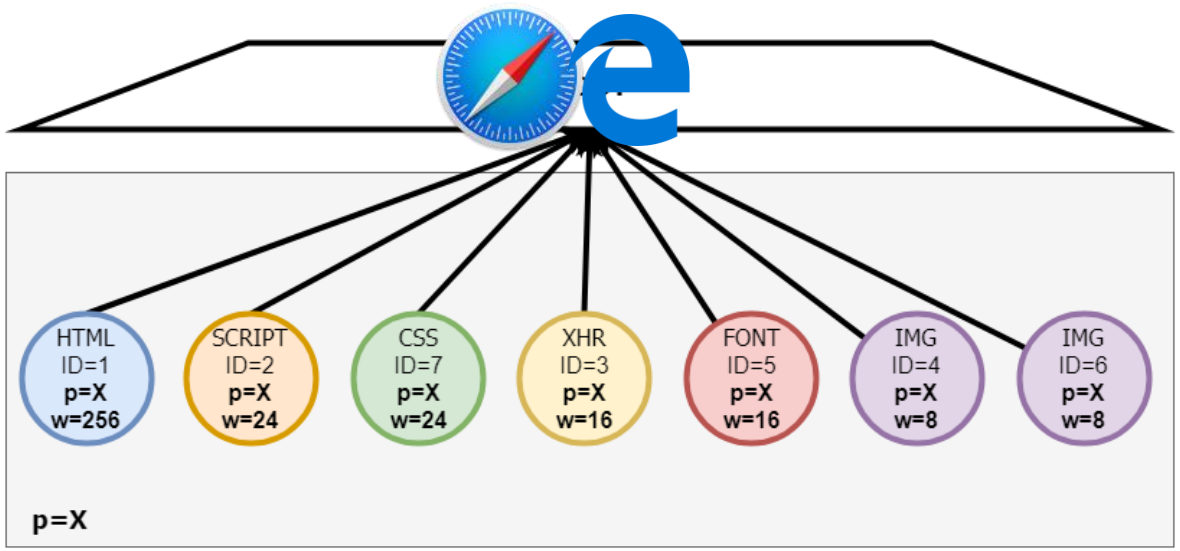
https://www.youtube.com/watch?v=ct5MvtmL1NM
https://github.com/andydavies/http2-prioritization-issues
https://www.slideshare.net/patrickmeenan/http2-in-practice
https://discourse.haproxy.org/t/http-2-prioritization/4578/3
https://twitter.com/bagder/status/1222143040577589248?s=20
https://medium.baqend.com/chromes-service-workers-break-http-2-priorities-649c4e0fa930

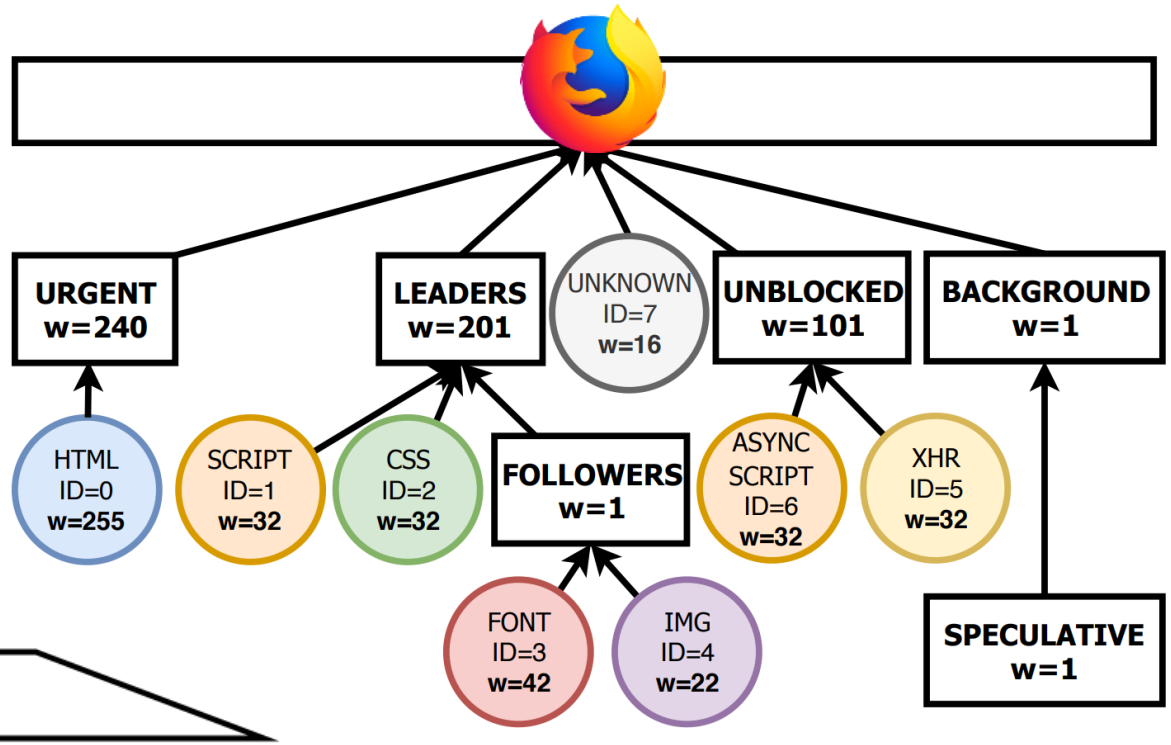
BUT: difficult to do a server-side override

Find best parent



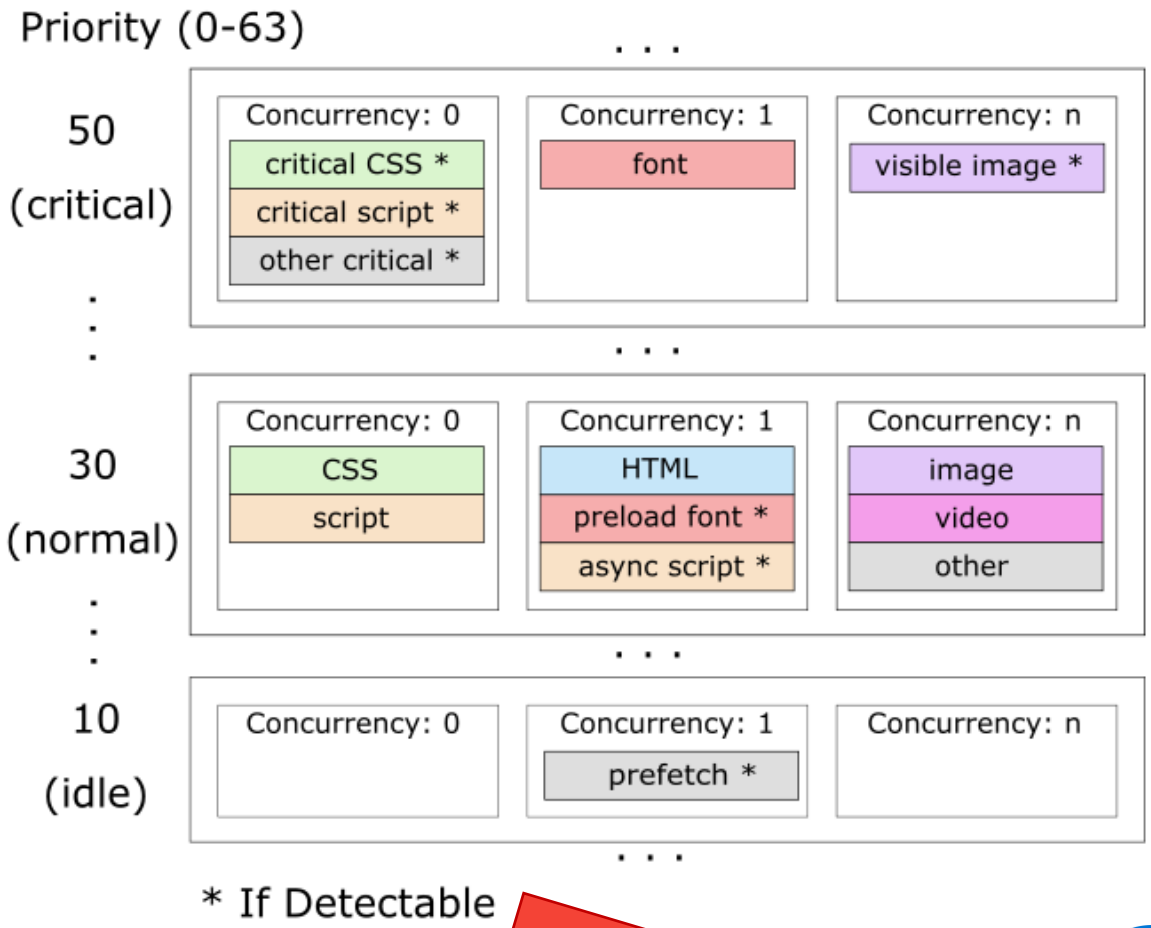
As sibling with high weight





As sibling under correct placeholder with some weight

Server-overrides ~= redo the whole thing



Look at browser's PRIORITY messages

- Guess which browser
- Put resources into fully new server-side scheme

not on , revert to mime-type

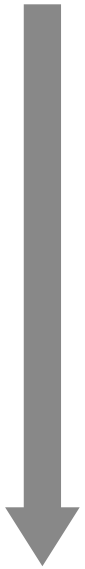
https://twitter.com/programmingart/status/1222550479168733186?s=20


THIS IS FINE.

Current HTTP/3 proposal under consideration: Back to SPDY basics

urgency

highest

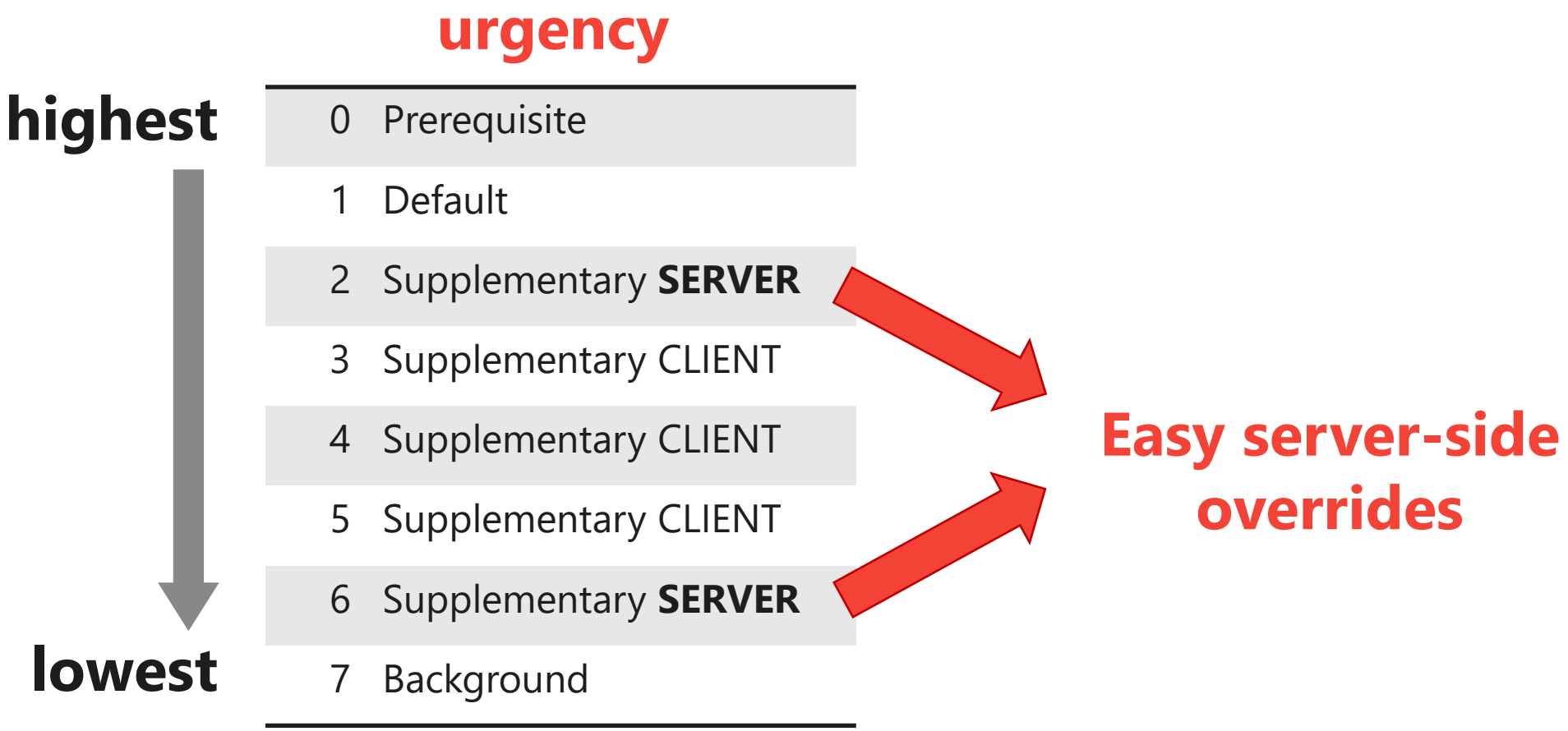


lowest

0	Prerequisite
1	Default
2	Supplementary **SERVER**
3	Supplementary CLIENT
4	Supplementary CLIENT
5	Supplementary CLIENT
6	Supplementary **SERVER**
7	Background

https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md

Current HTTP/3 proposal under consideration: Back to SPDY basics



https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md

Current HTTP/3 proposal under consideration: Back to SPDY basics

highest

lowest

urgency	
0	Prerequisite
1	Default
2	Supplementary **SERVER**
3	Supplementary CLIENT
4	Supplementary CLIENT
5	Supplementary CLIENT
6	Supplementary **SERVER**
7	Background

incremental	
0	Sequential
1	Can be Round-Robined

Easy server-side overrides

https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md

Current HTTP/3 proposal under consideration: Back to SPDY basics

```
:method = GET
:scheme = https
:authority = example.net
:path = /menu.png
priority = u=4, i=?1
```

Using an HTTP header:

- Easy to debug
- Easy to track
- Easy to reason about

```
fetch("image.jpg", {
    headers: {
        "priority": "u=4, i=?1"
    }
});
```

https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md

Many open questions…

How to handle round-robin in practice?

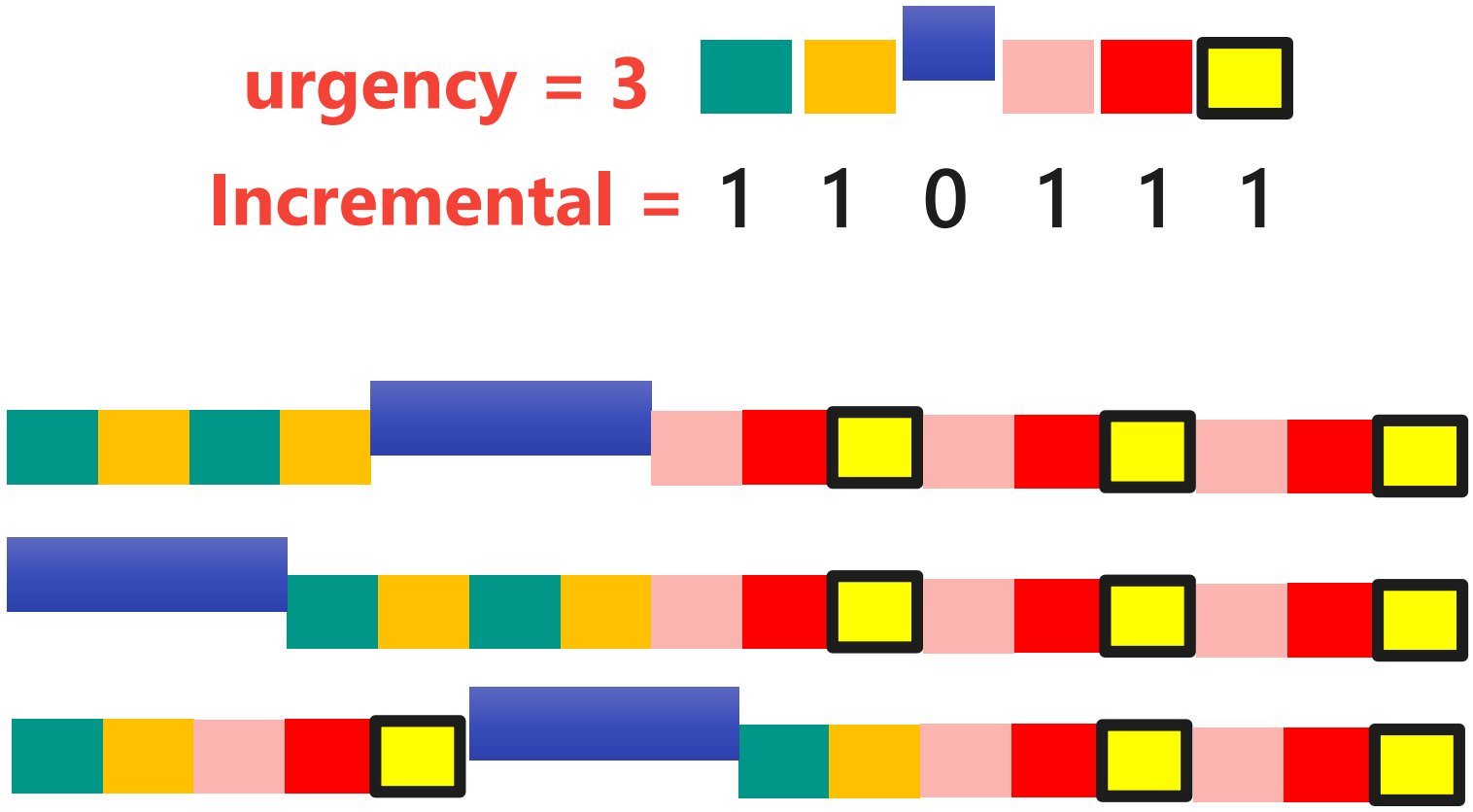
urgency = 3

Incremental = 1 1 0 1 1 1

https://github.com/kazuho/draft-kazuho-httpbis-priority/issues/94
https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md
https://github.com/httpwg/http-extensions/issues?q=is%3Aissue+is%3Aopen+label%3Apriorities

Many open questions…

How to handle round-robin in practice?



Note: unfair Round-Robin no longer really possible

https://github.com/kazuho/draft-kazuho-httpbis-priority/issues/94
https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md
https://github.com/httpwg/http-extensions/issues?q=is%3Aissue+is%3Aopen+label%3Apriorities

Many open questions…

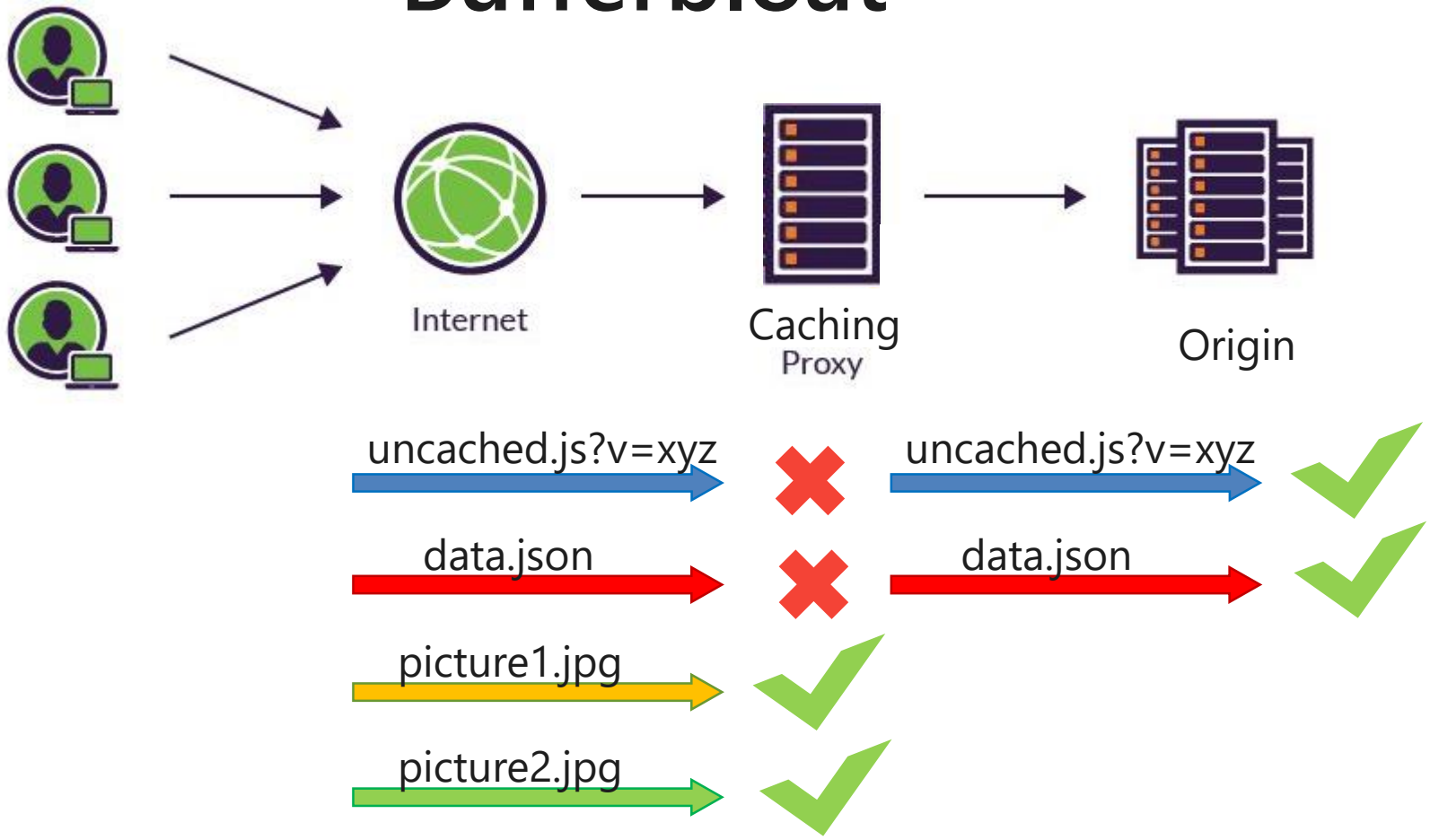
- Many people don’t like using HTTP headers for this
- Headers also cannot be used for changing priorities
- Exposing this to JavaScript had lots of push-back
- Are 8 levels enough? Do they really need semantics?
- What about the fairness issue?

https://github.com/httpwg/http-extensions/blob/master/draft-ietf-httpbis-priority.md
https://github.com/httpwg/http-extensions/issues?q=is%3Aissue+is%3Aopen+label%3Apriorities

Problem 3:

There is more than one protocol layer

Bufferbloat

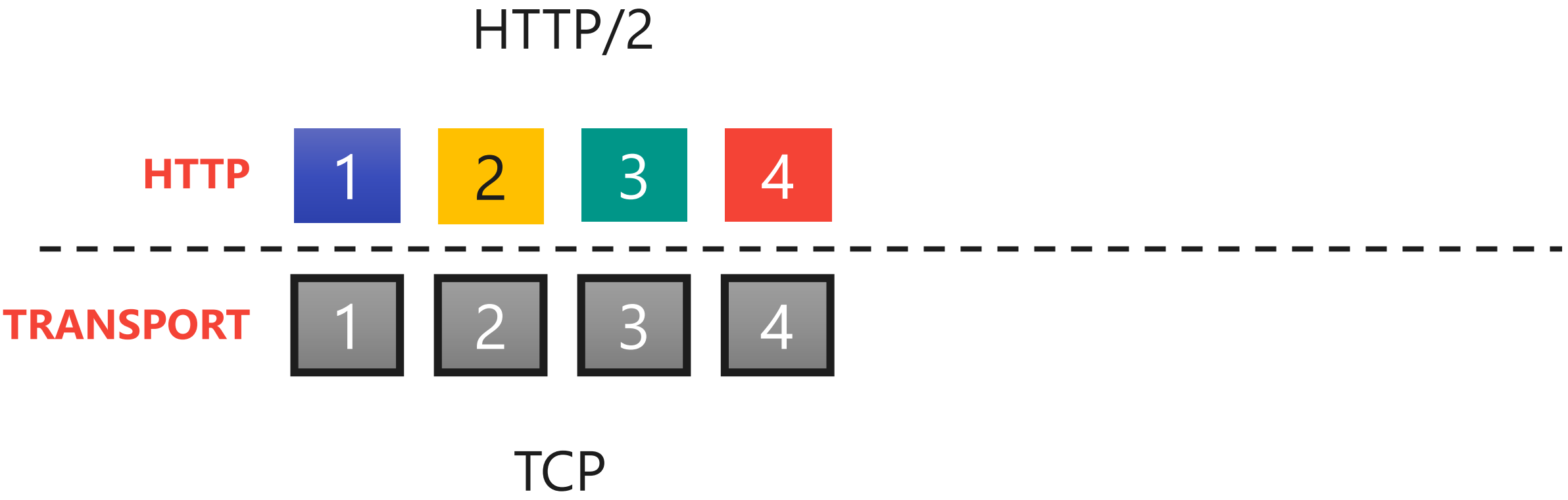


Expected:

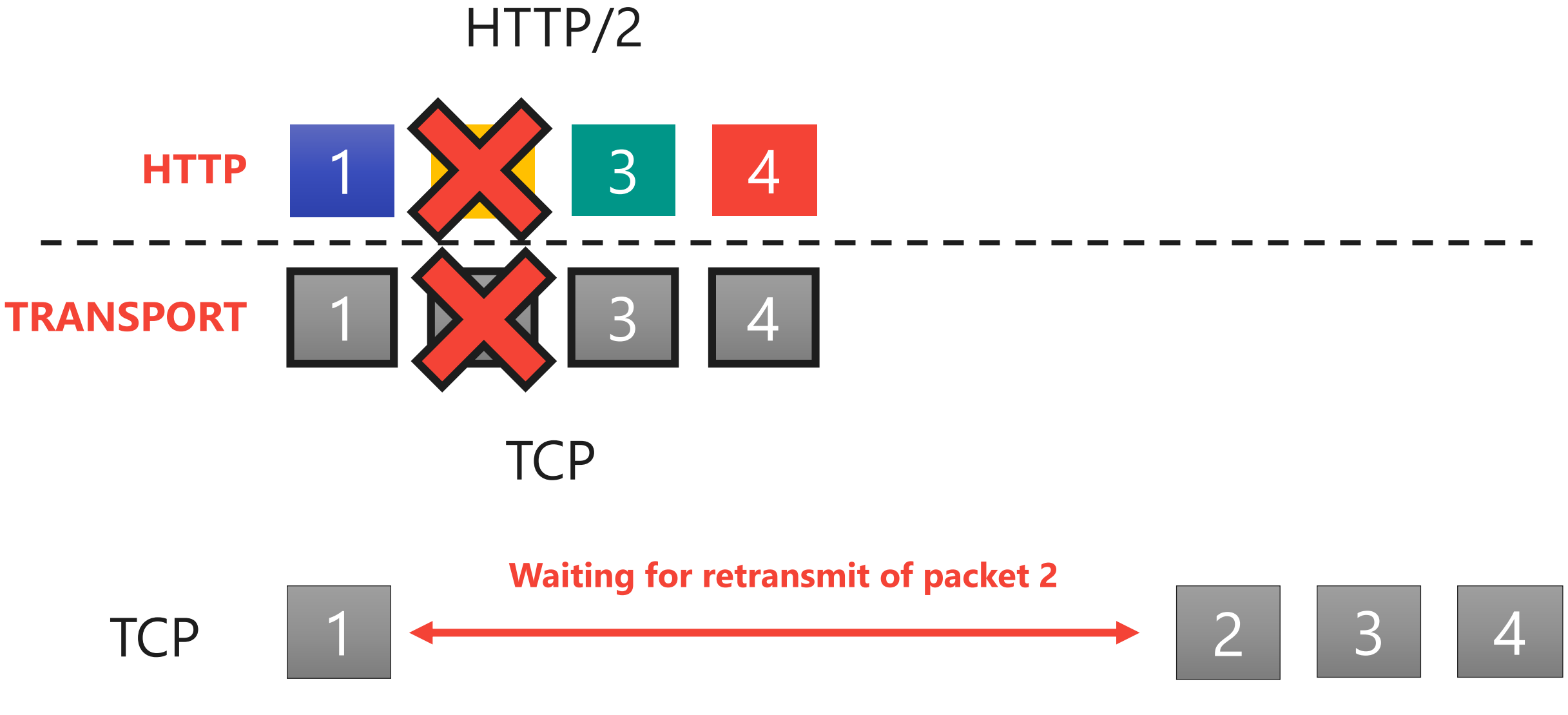
Actual:

https://www.youtube.com/watch?v=pq_xk_Pecu4
https://www.youtube.com/watch?v=ct5MvtmL1NM

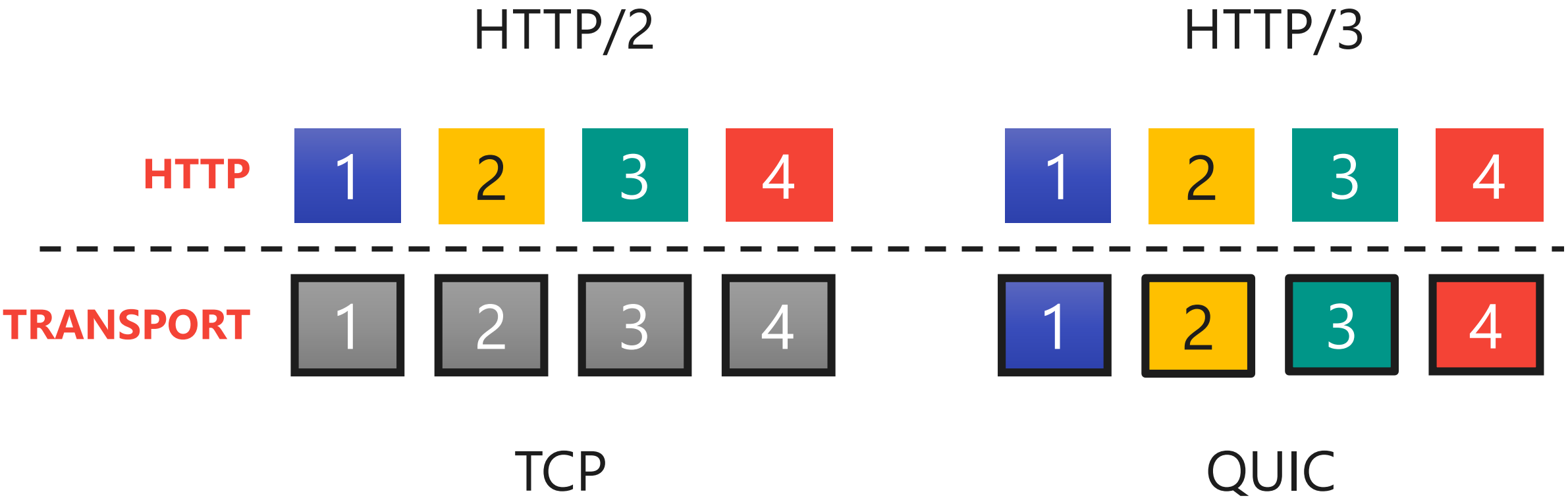
TCP Head-of-Line blocking



TCP Head-of-Line blocking

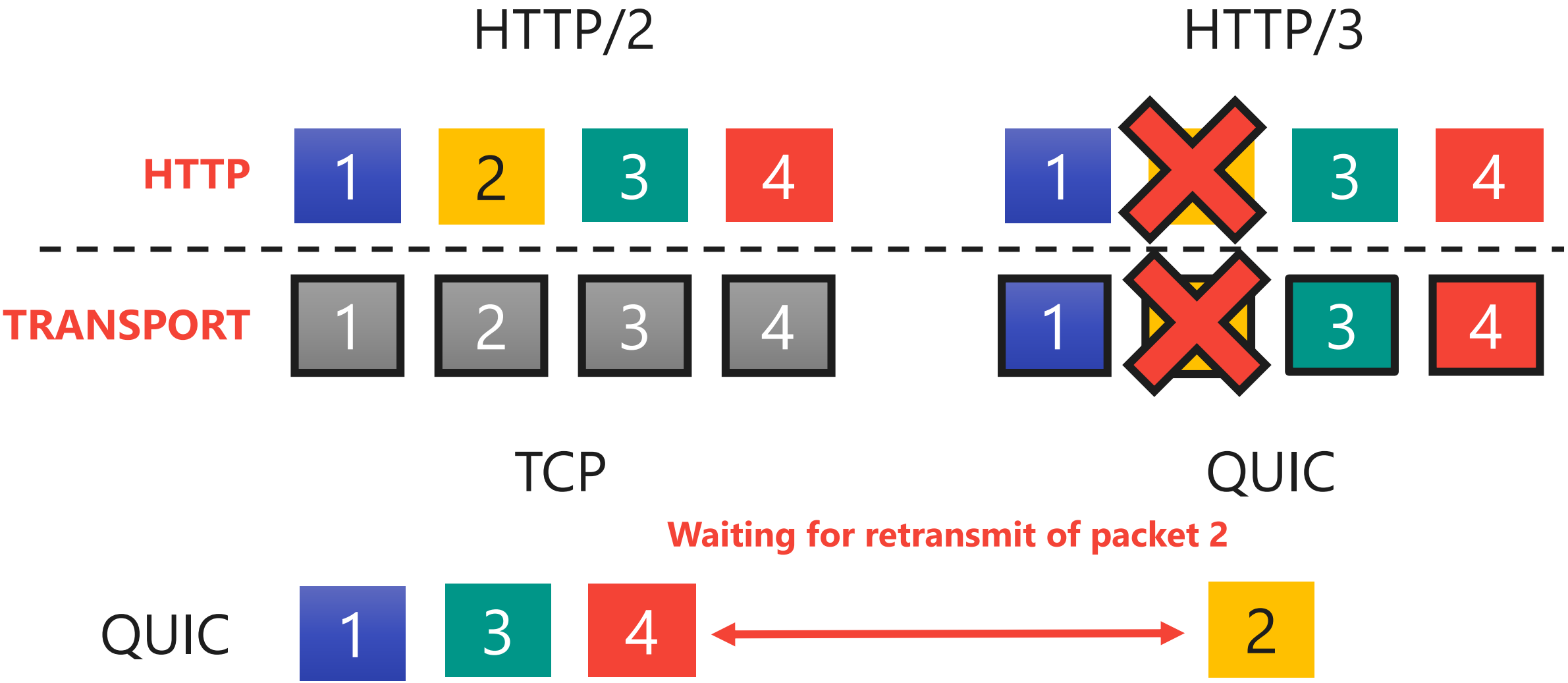


QUIC has no more Head-of-Line blocking!



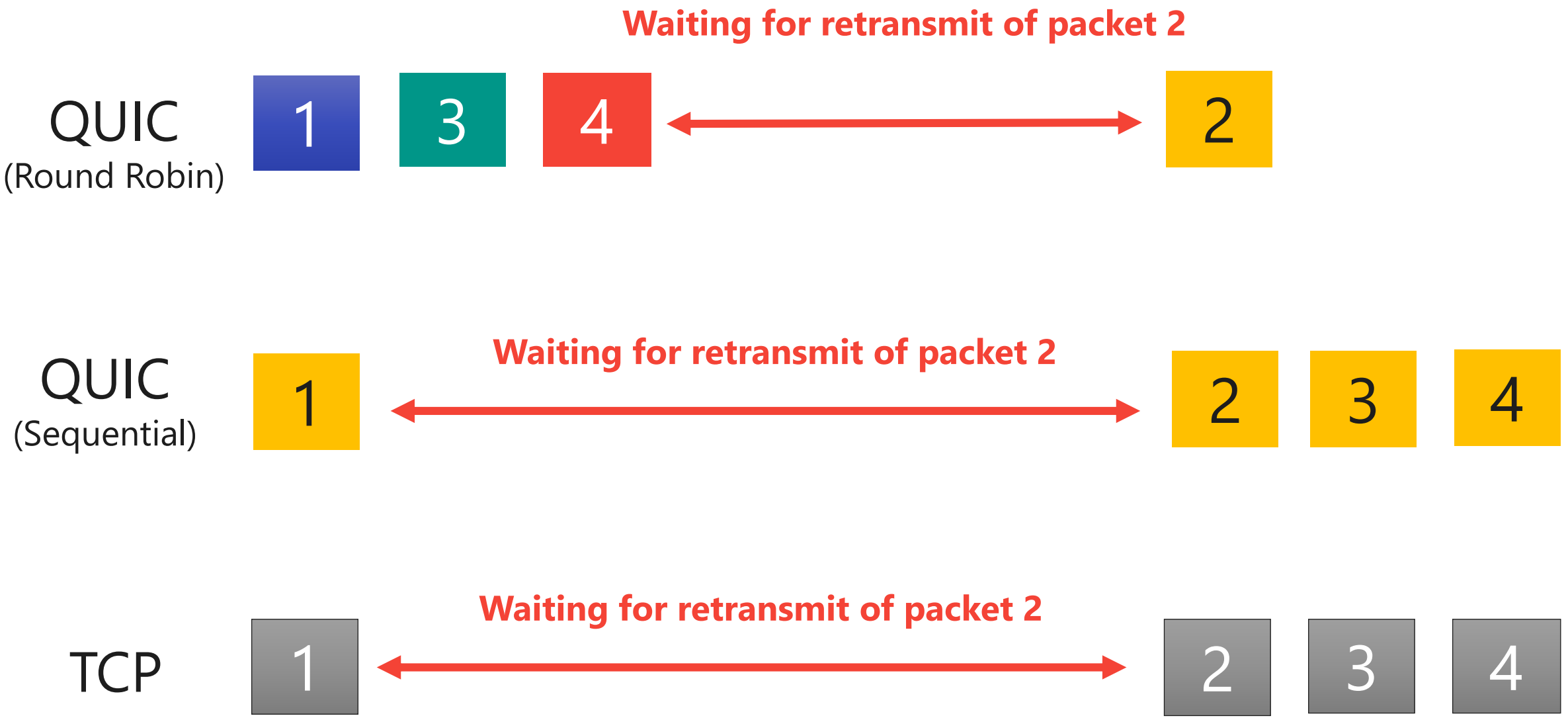
https://h3.edm.uhasselt.be/

QUIC has no more Head-of-Line blocking!



Has QUIC really solved Head-of-Line blocking?

Has QUIC really solved Head-of-Line blocking?



Multiple other challenges in QUIC

Retransmits, multipath, flow control, scheduling APIs,…

Please read our paper:

https://bit.ly/quicH3

(no one else will)

Problem 4:

There is no problem

Incompatible results

Edge 50% slower

9/34 deployments broken

Max 3.1% difference

Haven’t seen large scale complaints…

Incompatible results

Edge 50% slower

9/34 deployments broken

Max 3.1% difference

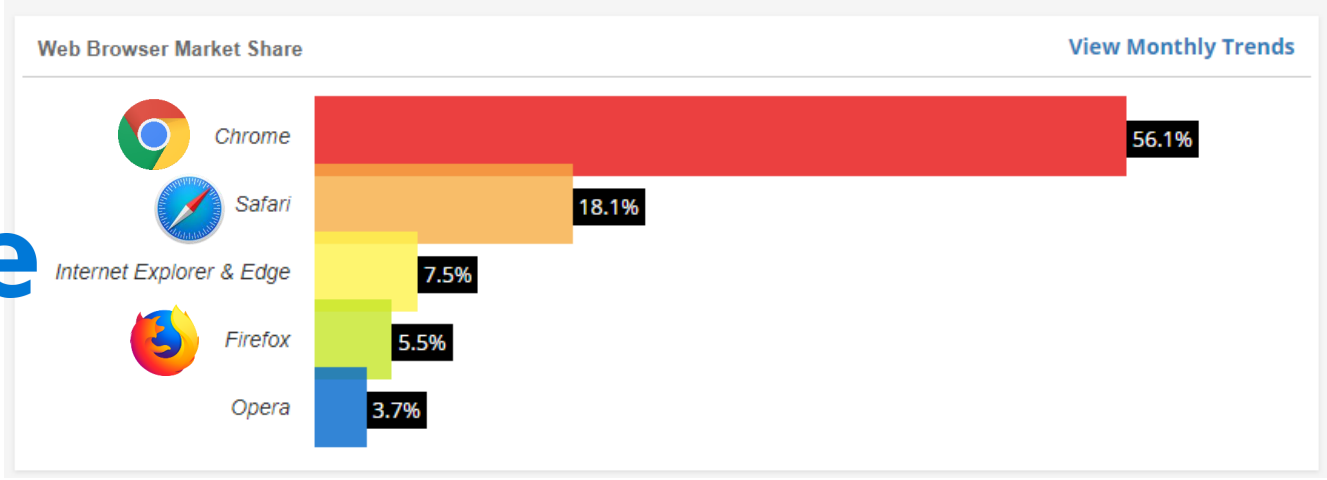
Haven’t seen large scale complaints…

Web Performance isn’t important

Robin has wasted his life

Personal conclusion

- Most important for complex pages on slow networks
- Network often isn't the bottleneck anymore… (single thread JS)
- If things break, they break hard, and are fixed
- People might not identify problems as being priority-related
- Uneven browser usage shares across platforms



https://www.w3counter.com/globalstats.php

Personal conclusion

- QUIC will highlight some long-standing and new issues
- Proposed HTTP/3 scheme isn't going to solve all problems
- But it will be easier to debug, understand, reason about
- It can also be retrofitted onto HTTP/2!!!

Problem 5:

Too many slides, not enough time

Browser doesn't know

1. Size of resource
2. If the resource can be used progressively
3. What the resource will actually do
4. If the resource references other resources
5. The "critical path"

So it has to guess

1. Mime-type
2. Position in the document
3. How it hopes developers use things like async, defer, preload, …

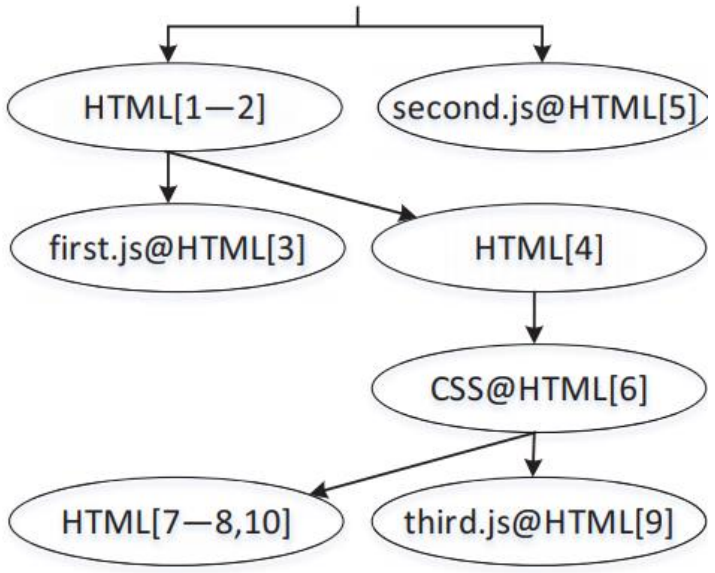
}

More than the server typically knows

Server could know…

```
1  <h1>Text</h1>
2  <p>Text</p>
3  <script src="first.js"/>
       <!--Reads <p> tag-->
4  <b>Text</b>
5  <script src="second.js"/>
       <!--Accesses no DOM nodes-->
       <!--or JS state from first.js----->
6  <link rel="stylesheet" href="...">
       <!--CSS-->
7  <span>Text</span>
8  <span>Text</span>
9  <script src="third.js"/>
       <!--Writes <b> tag-->
10 <span>Text</span>
```

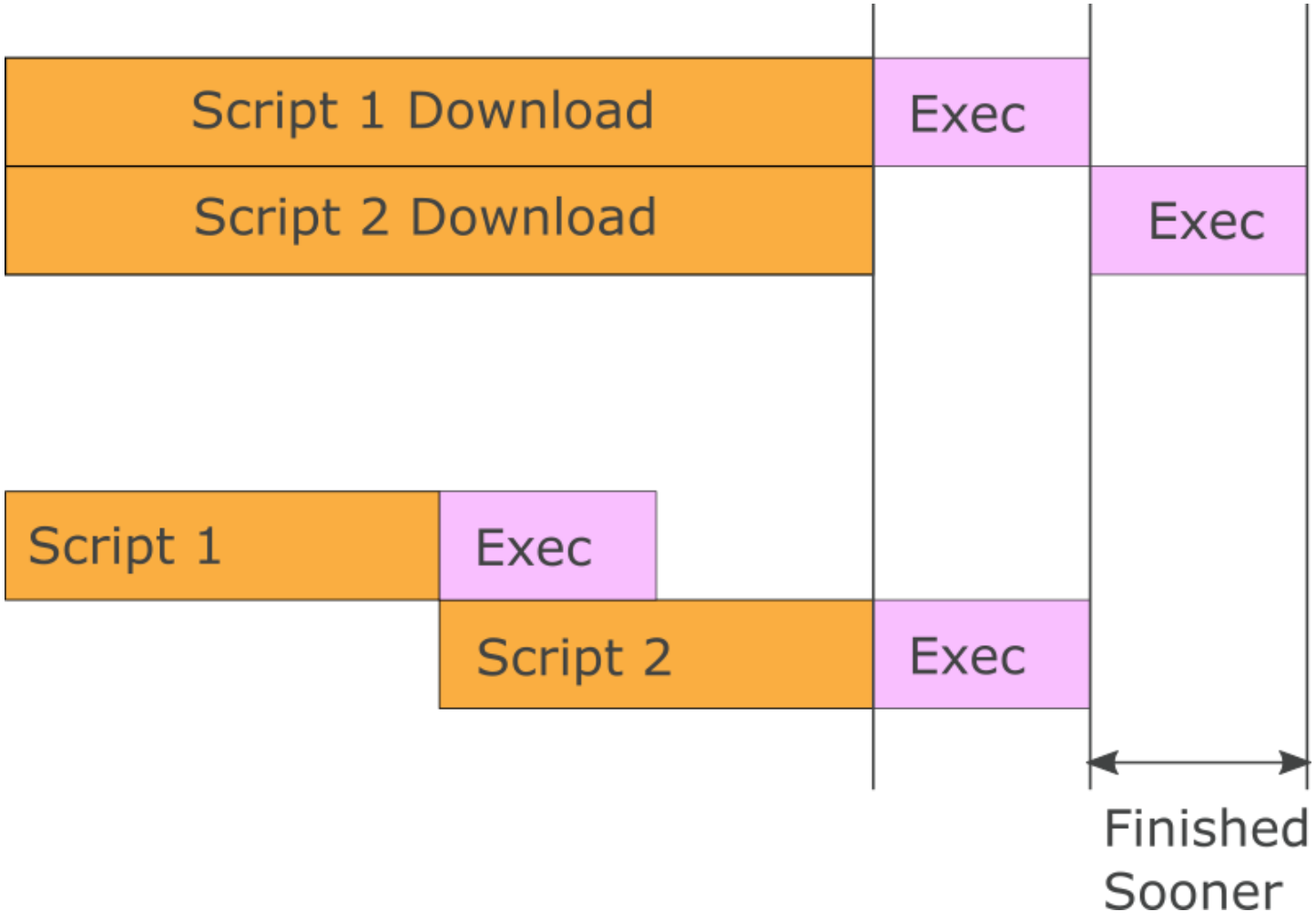
(a) The HTML for a simple page.



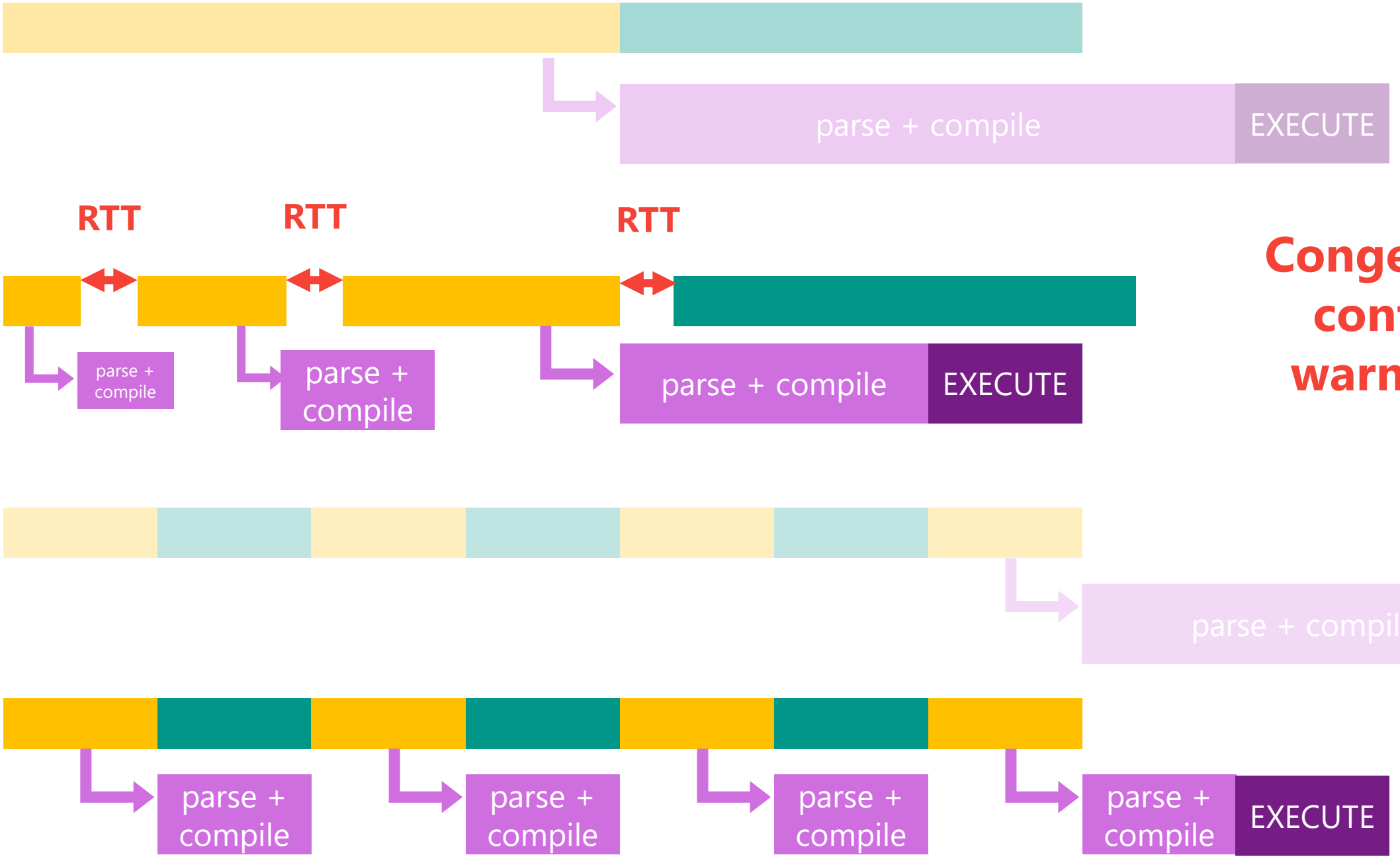
(b) The dependency graph generated by Scout.

http://web.mit.edu/polaris/
https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final177.pdf
https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-wang-xiao-sophia.pdf

Another aspect: pipelined executions on main thread



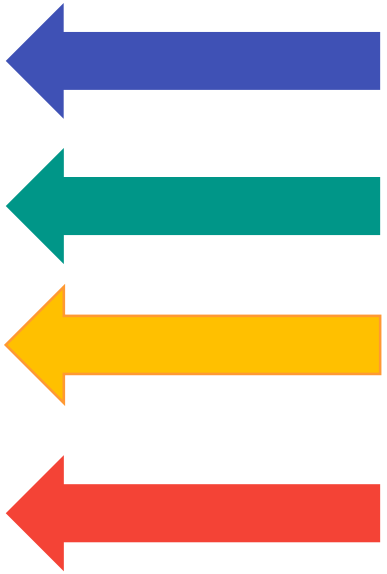
https://blog.cloudflare.com/better-http-2-prioritization-for-a-faster-web/



https://v8.dev/blog/v8-release-78
https://medium.com/reloading/javascript-start-up-performance-69200f43b201

Is Round-Robin bad?

HTTP/1.1 (TCP)



HTTP/2 (TCP)

HTTP/3 (QUIC)

Which resources can/should be Round-Robined?

HTML : can : if not too big

JS : partly : not blocking, rest could be ok

CSS : no : not blocking, rest could be ok

Video: no : please don’t

Fonts : no : please don’t

Images: some : useful for size metrics (but then: use CSS / <img> attributes please) but generally serious doubts about usefulness progressive JPEGs (see next slide)

On the need for progressive JPEGs

⇒ Main downside in chrome is not taking advantage of progressive jpgs… personal opinion: don't matter that much

https://tobias.is/blogging/even-faster-images-using-http2-and-progressive-jpegs/

http://blog.patrickmeenan.com/2013/06/progressive-jpegs-ftw.html

https://blog.radware.com/applicationdelivery/wpo/2014/09/progressive-image-rendering-good-evil/ : progressive JPEG is worse than normal…

https://twitter.com/kornelski/status/1222618103873441793?s=20 : limited proof it works

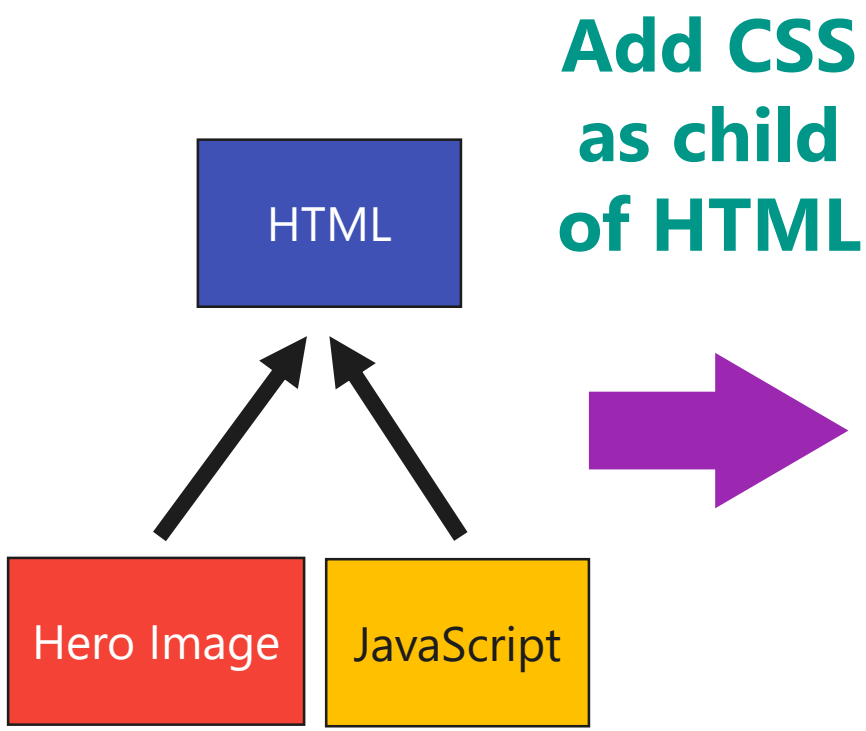
⇒ https://www.davidtnaylor.com/eyeorg.pdf : speedindex is badly correlated with user opinions (See also https://www.mediawiki.org/wiki/Wikimedia_Performance_Team/Perceived_Performance: "Existing RUM metrics like onload, TTFP as well as SpeedIndex correlate very poorly with user-perceived page load time")

⇒ Maybe will change in future with AV1, but…

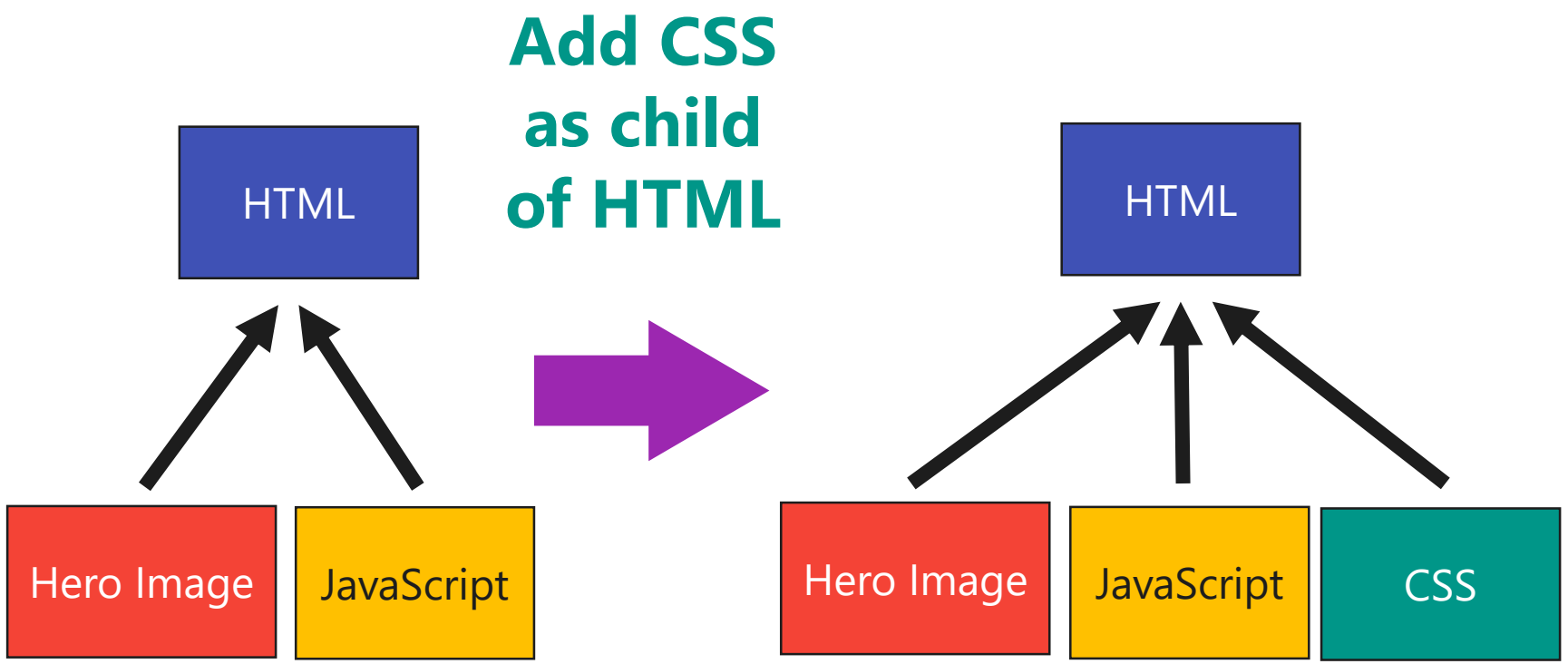
⇒ Also: webp is smaller, so what matters more: file size or blocky previews?

Preloadscanner: doesn't always know an image will be off-screen at that point -> will have to change priorities down the line (Chrome does do this)

2 PRIORITY Messages Between Client and Server

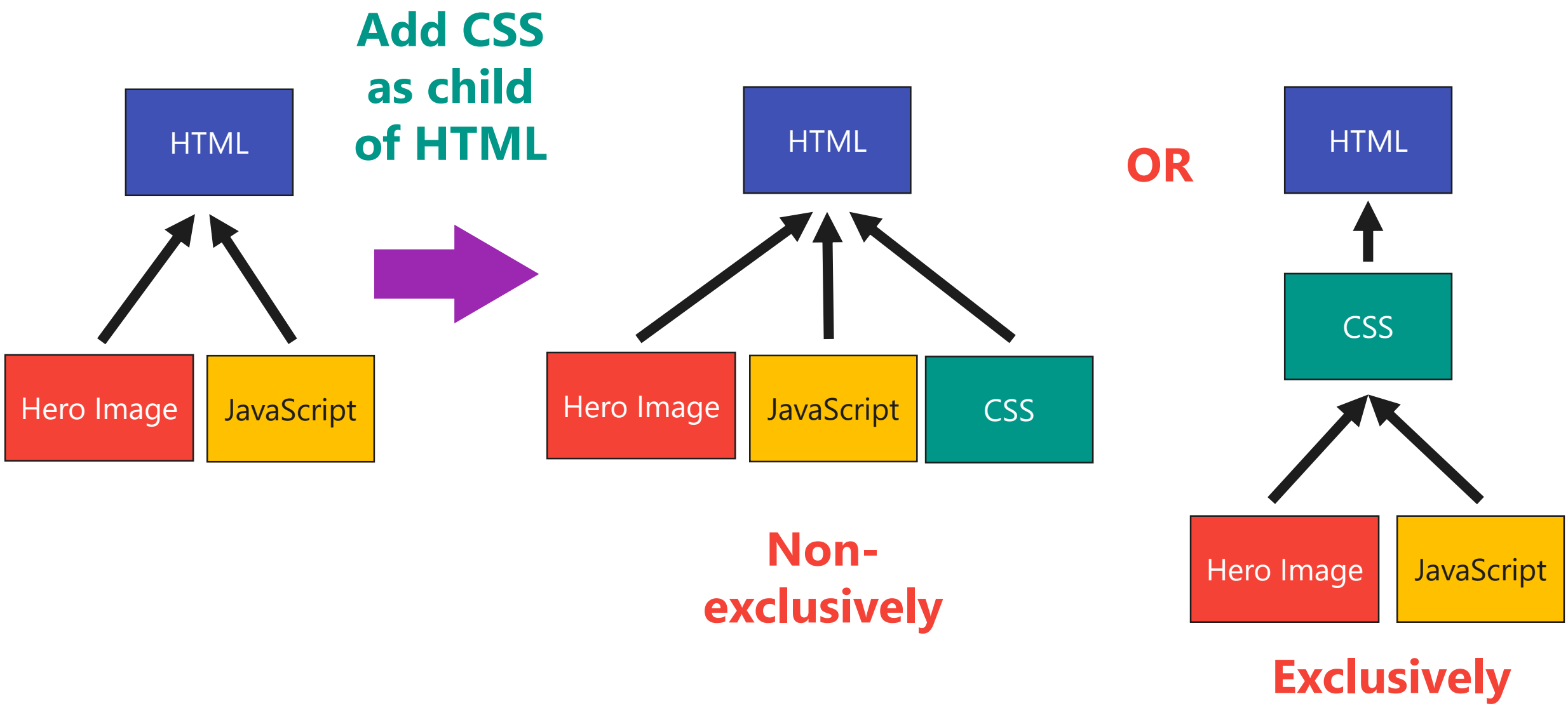


2 PRIORITY Messages Between Client and Server

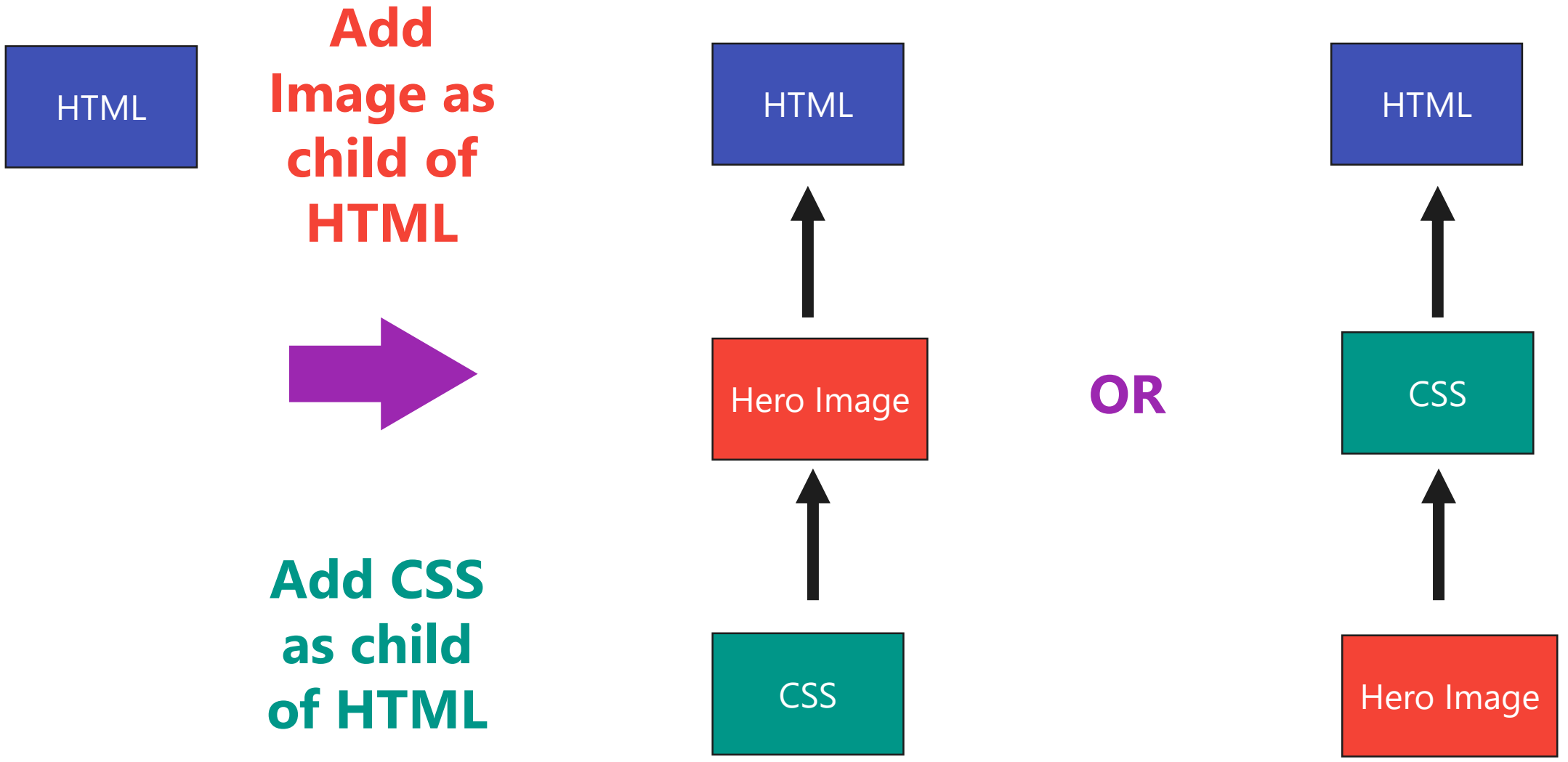


Non-exclusively

2 PRIORITY Messages Between Client and Server

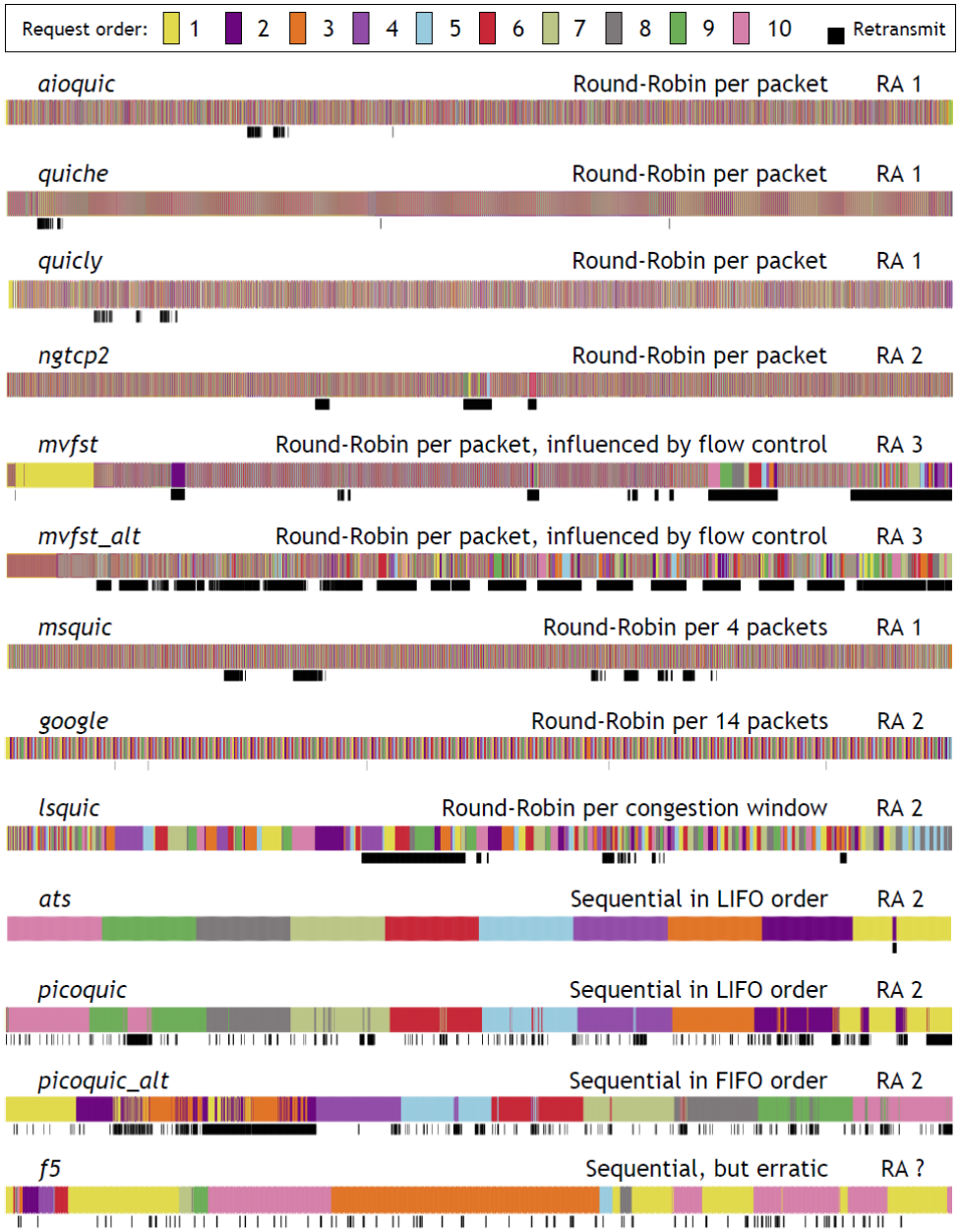


HTTP/3 : PRIORITY messages should be ordered...



https://h3.edm.uhasselt.be/files/HTTP3_Prioritization_extended_3jul2019.pdf

QUIC: introduces new problems: retransmission!



Welcome to the jungle

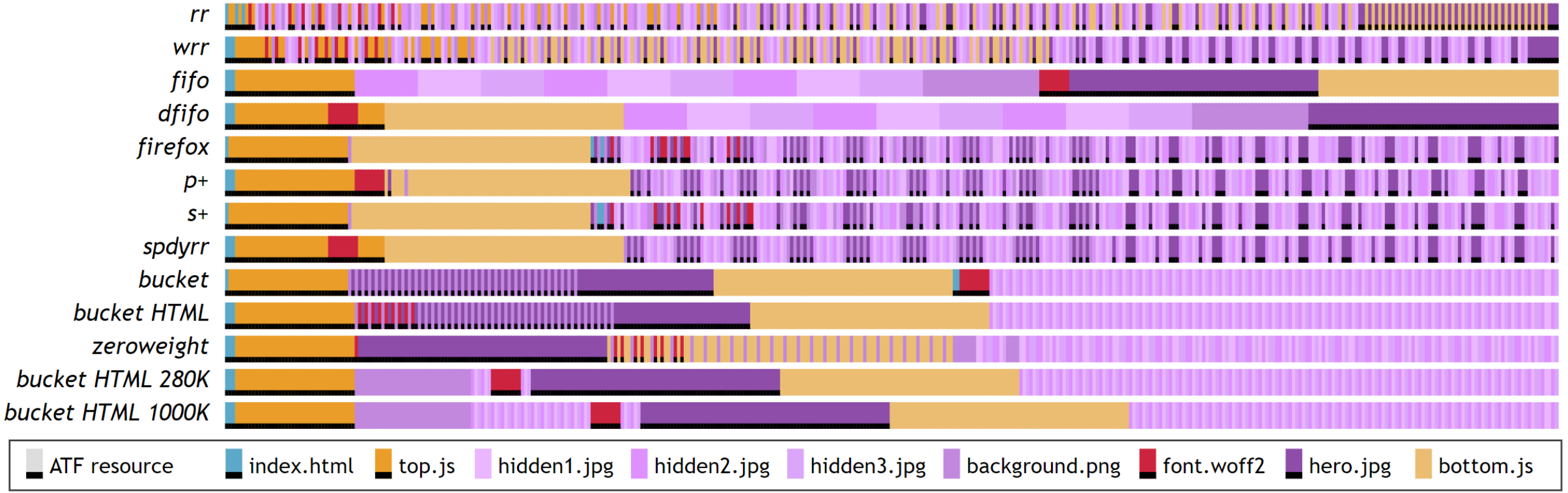


Image sources

- https://teurastamo.com/wp-content/uploads/2015/10/blackfood_burger.png
- https://upload.wikimedia.org/wikipedia/commons/e/ea/Oreo_biscuits_%28transparent_background%29.png
- https://pngimg.com/uploads/m_m/m_m_PNG10.png
- https://hellolipa.com/wp-content/uploads/2018/10/ZBP5.jpg
- https://www.walmart.ca/en/ip/broccoli-stalks/6000016950304
- https://media.istockphoto.com/photos/vegetables-carrot-isolated-on-white-background-picture-id519684644?k=6&m=519684644&s=612x612&w=0&h=lILeiVzaD-3uxvPxaAP5myTsFGURIavCRaBT-tZX99Y=
- https://thumbs.dreamstime.com/b/big-hamburger-white-background-big-hamburger-white-background-close-up-99672617.jpg
- https://media.istockphoto.com/photos/potatoes-fries-isolated-on-white-background-fast-food-picture-id496313136?k=6&m=496313136&s=612x612&w=0&h=_IUh7QgCF2nzIb4_5-TSz3D9L-mT1fZtFwRkbC4BCTI=
- https://image.shutterstock.com/image-photo/green-chicks-carrots-boxes-shadow-260nw-672226147.jpg
- http://www.barrestaurant-devrienden.nl/wp-content/uploads/2016/01/12.jpg
- https://scontent-cdt1-1.cdninstagram.com/v/t51.2885-15/e35/66218838_942468512777551_4547401241838710995_n.jpg
- https://www.unilad.co.uk/wp-content/uploads/2015/10/UNILAD-question-mark-mystery-meat36659-584x426.jpg
- https://s3.amazonaws.com/wowfilters/content-images/the-three-mouths-instagram-filter-1.jpg
- https://pnghunter.com/search/waiter
- https://www.stickpng.com/assets/images/580b57fcd9996e24bc43c567.png
- https://2.bp.blogspot.com/-C3TpcrQ0s20/UW7zBvLq2SI/AAAAAAAAAbA/SgKjXVHLGVE/s1600/Alphabet+Soup.png
- https://lh3.googleusercontent.com/proxy/x749arvZG1YKUSZ1UeH_Yw6wHOnW9IxahES5ITmMiXKWRU-Zc-uKr4mk4yZKfio-cqKl5vV4GyJfSNckyApqy9r6Jco-HQ72wHCXp2LmsR1RqMZpyF78RBCf0AhlsCjpb5VGDa4

Image sources

- https://i.pinimg.com/originals/99/e9/68/99e9689aa6a7a68d846ae942b401eacd.jpg
- https://2.bp.blogspot.com/-7ED7iessIXU/WmLu04G6HWI/AAAAAAAAKec/yslLVyakN-cL2D0CKD23IZUEUsdIFa13QCLcBGAs/s1600/Chrome_win_browser_wars.png
- https://www.waffleworkshop.com/waffle-workshop6.jpg
-

Glide

by SLIDOR

A free PowerPoint Template made by Slidor.

VISIT US